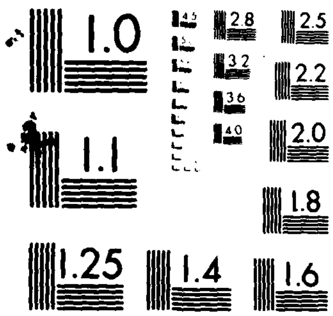
AD-A108 744 NAVAL TRAINING EQUIPMENT CENTER ORLANDO FL F/G 9/2
AN ANALYSIS OF MICROCOMPUTER TECHNOLOGY FOR APPLICATION TO REAL--ETC(U)
JUL 79 C F SUMMER, G A WYNDLE
UNCLASSIFIED NAVTRAEQUIPC-IH-313 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Technical Report [illegible]

AN ANALYSIS OF [illegible]
FOR APPLICATION TO REAL-TIME [illegible]

CHARLES E. SUMMER, P.E.
GERALD A. WENDLE
COMPUTER SYSTEMS LABORATORY

JULY 1979

Report for Period July 1976 through April 19[illegible]

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA [illegible]

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NAVTRAEQUIPC[illegible] IH-313

2. GOVT ACCESSION NO.
AD-A108 744

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
An Analysis of Microcomputer Technology For Application To Real-Time Trainers

5. TYPE OF REPORT & PERIOD COVERED
July 1976 thru Apr 1979

6. PERFORMING ORG. REPORT NUMBER
NAVTRAEQUIPCEN IH-313

7. AUTHOR(s)
Charles F. Summer, P.E.
Gerald A. Wyndle

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Computer Systems Laboratory
Naval Training Equipment Center

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Project 5741

11. CONTROLLING OFFICE NAME AND ADDRESS
Commanding Officer
Naval Training Equipment Center
Orlando, FL 32813

12. REPORT DATE
July 1979

13. NUMBER OF PAGES
113

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

computers	real-time systems	firmware
microcomputers	life cycle costs	microcode
microprocessors	VLSI modules	software partitioning
computer architecture	bit-slice technology	
microprogramming	software	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Conceptual approaches to introducing advanced microcomputer technology in real-time trainers were investigated. Four types of computer system architectural approaches were examined and eliminated for various reasons. Eight available microcomputer/microprocessor modular families were also analyzed for trainer application. A conventional general purpose system was synthesized and used as a baseline for comparison. A unique functionally modular multiple microcomputer architecture is described that will satisfy the processing,

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED 390463
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract (Continued): computation and control requirements of modern high performance trainers. A life cycle cost model is described and the life cycle cost of the conventional GP approach is compared with the life cycle cost of the multiple microcomputer approach. Technologies required to achieve the unique architecture are summarized and a hardware-firmware (microcode)-software system control algorithm is described in conceptual terms.

Future R&D effort required to develop the control algorithm is identified.

Acces[illegible]
NTIS
DT[illegible]

A

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

Over the past decade, the proliferation of computers embedded in trainer systems with the attendant proliferation of assembly-level languages has resulted in increased life-cycle costs of all types of real-time trainers. The advent of LSI and VLSI technology embodied in state-of-the-art microprocessors and microcomputers has suggested computer system architectural concepts that potentially can reverse this proliferation. Furthermore, these concepts offer the possibility for extensive standardization, modularity and performance improvements that significantly impact and reduce life-cycle costs of real-time trainers.

Various conceptual approaches to introducing advanced microcomputer technology in real-time trainers have been under investigation at NAVTRAEQUIPCEN for over two years. The question of how this technology can be properly and effectively introduced in the design of real-time trainers has been addressed during this investigation. Results of the investigation are documented in this report. They indicate the technical and cost feasibility of applying microcomputer technology in a functionally dedicated multiple configuration to handle the processing, control, computation and other simulation functions of real-time trainers. Currently embedded conventional general purpose computers are used to satisfy such requirements.

Basic cost tradeoffs were made between an F-18 OFT trainer concept using commercially-available computer equipments and the same trainer concept employing multiple microcomputers formulated with bit-sliced LSI modules.

The analysis also identifies the basic technologies required to achieve a working multiple microcomputer system concept as described in this report.

SECTION II

STATEMENT OF PROBLEM

BACKGROUND AND DESCRIPTION

The ability to improve cost effectiveness in training by simulation (i.e., trainer systems) is limited in an important technical area of trainer design and field support. This is the design of the computer system considering both hardware and software and field support in terms of lack of commonality in the areas of computer spares, maintenance personnel training, hardware documentation, software documentation, programmer experience, knowledge and training.

Modern trainers (e.g., aircraft operational flight trainers (OFT's), weapon system trainers (WST's), cockpit procedures trainers (CPT's), sonar trainers, EW trainers, shipboard trainers, submarine trainers, tactics trainers, and the like) universally employ embedded commercially available digital computers of various types and designs to perform the real-time control, processing and computation required for the simulation. Application of commercial computers has in the recent past provided the most cost effective trainer design approach since no unique computer hardware development was required. Reasonable delivery schedules for trainers have been afforded by availability of such "off-the-shelf" computers. Standard military computers have been employed only to a very limited extent because of their performance limitations and high initial cost when employed in real-time simulators for training.

Under current procurement policies and practices, the embedded computing equipment must be selected for each trainer system design and application by the respective system contractors. The selection criteria is based on NAVTRAEQUIPCEN specifications and the system contractor's analysis of the requirements and available computer hardware. To date, these computers have been programmed almost exclusively in the assembly language of the selected computer. The software package developed and delivered with each trainer is unique to that trainer and to the language of the computer involved. As a result of this situation, there is little or no commonality of computer hardware, no commonality of computer languages, and no commonality of computer software programs even between trainers of a similar class (e.g., OFT's, or CPT's, or WST's).

This proliferation of different computer hardware types (which also results in the proliferation of different language types) has resulted in a Navy training device inventory (or under contract) as of March 1979 of 793 digital computers comprised of 63 different hardware types that require a knowledge (for programming, documentation, etc.) of 43 different assembly level languages. Of the 793 computers cited above, a total of only 64 are

standard military Navy units of 13 different types. This large inventory of different types of computers requires extensive and costly life cycle field support in terms of large numbers of unique spare components for hardware maintenance, the training of many maintenance personnel on many different types of computer hardware, unique hardware documentation for each type, unique software documentation for each type, unique software maintenance and unique programmer training and indoctrination in the many computer languages over the life cycle of the various trainers.

Although the acquisition cost of computer hardware applied to trainers has been reduced dramatically over the past three years, the acquisition cost of the computer software has significantly increased over the same period. It is currently estimated to range from 3:1 to as high as 10:1. In 1974 a brief NAVTRAEQUIPCEN study indicated the average was at least 2.5:1. That study was based on only five types of trainers. Other data generally available in industry on 10 year life cycle maintenance costs of software indicates such costs can easily increase the software/hardware dollar ratio of 3 to 1 or more.

A major contributing factor to the high software acquisition and field support costs is the fact that programming is extensive, coding is performed in the assembly language of the computers involved and there is no software commonality between similar trainer types. There is currently no language commonality or compatibility among different types of computers embedded in trainers. Since each new program is in a different language, documentation costs are high and there is no ability to apply (or transfer) program modules developed for one trainer to another similar trainer regardless of application and/or mathematical similarity between them. The field support costs include trainer software maintenance, software design changes and software updating as well as the continual training of programming personnel of the software support organizations in many different computer languages. Turnover of programming and maintenance personnel further accentuate these problems and increase support costs over the life cycle.

System architectures designed with commercially available computers and/or standard military computers (e.g., AN/UYK-7, AN/UYK-20, etc.) have not been optimum in many cases in terms of real-time processing, computation and control performance required by modern high performance trainers and/or for realistic life cycle cost effectiveness. The lack of suitable computer hardware modularity has resulted in performance overkills (excess capabilities) for some trainer computers and underkills (deficiencies) in many others. In most instances, multi-processor configurations are required to solve the processing deficiencies of a single computer. In addition to the fact that these system architectures are more costly to implement, the ability to program, synchronize and control such configurations in real-time is very complicated and contributes to the increased software costs. This is a direct result of the computing equipment not being initially conceived and designed to be organized and efficiently applied in system configurations required for real-time trainers.

The availability of a suitable and standard high order language (HOL) would decrease both life cycle costs and technical software programming problems. However, the only standard HOL universally available with most cost effective commercial computers suitable for trainer applications is ANSI FORTRAN IV. The availability of FORTRAN IV compilers that can generate optimized and efficient object code capable of being used in most real-time trainers is also significantly limited.[1] As of the time of this report preparation (1979) there are only two or perhaps three efficient FORTRAN compilers available with contemporary commercial computers suitable for real-time trainers that can be used a pseudo (or de facto) standard programming language. The use of FORTRAN IV is much less than optimum as a standard source language for real-time trainers. The power of FORTRAN is in its numerical computation and processing constructs. However, on the average, this represents only approximately twenty to twenty-five percent of the total computer workload for the usual A/C OFT (as an example). The remaining seventy-five to eighty percent is involved in extensive bit, byte, logical, input-output, handling data structures, interrupt and special control operations. The average FORTRAN is significantly deficient in its ability to express these types of programming functions in its syntax and language constructs, and to optimally compile such source statements into object code for efficient real-time execution. Further, adequate run-time facilities (e.g., real-time operating systems) for FORTRAN are characteristically lacking. This also is a significant limitation on the ability to port (transfer) FORTRAN source code between various similar trainers.

Any approach to solving the problems of (1) continued proliferation of types of computer hardware, (2) continued proliferation of different computer languages, (3) reducing life cycle costs and (4) improving computer system performance in trainers must be considered in a total system context. The cost reduction benefits to be gained by using a standard or common HOL cannot be fully exploited if the compatible computer hardware is not modularly expandable and does not possess acceptable real-time processing capabilities, both at a cost effective price and available from multiple sources.

The advent of large scale integration (LSI) and very large scale integration (VLSI) technology embodied in state-of-the-art microprocessors and microcomputers suggests computer system architectural concepts that potentially can reverse the problems of non-standard computer equipment and language proliferation. Further, some concepts offer the possibility for extensive modularity and performance improvements that can significantly impact performance and reduce life cycle cost of real-time trainer systems.

1. The Efficiency of FORTRAN in Simulation Computers, F.A. Sigmund, Goodyear Aerospace Corp., Proceedings, 10th NTEC/Industry Conference, NAVTRAEQUIPCEN IH-294, Nov 1977.

Microcomputer system architectural concepts for trainers have been under investigation and analysis at NAVTRAEQUIPCEN for over two years. This report describes the investigative approach and analysis results of a concept of multiple microcomputers employed in a dedicated functionally-modular system architecture. The concept provides an approach to solving the previously mentioned problem areas and will eventually achieve the following:

a. Commonality of LSI and VLSI microcomputer family modules available from multiple commercial sources.

b. Commonality of computer hardware documentation between similar trainer types.

c. Processing performance improvement via intrinsic parallelism of the concept and optimized standard code.

d. Modular hardware structure can support tailored processing requirements for a wide spectrum of trainer types.

e. Overall trainer program is modular-partitioned for dedicated assignment of module to a specific microcomputer.

f. Modular-partitioning of the total program permits future standardization of individual program modules (routines) and commonality of software documentation.

g. Ability to accept future advances in VLSI microcomputer technology without early obsolescence and loss of software base via the mechanism of microprogramming.

h. Commonality of maintenance training to reduce life cycle costs.

i. Commonality of programmer training and orientation to reduce life cycle costs.

j. Concept can readily accept future application of the new DoD standard HOL (ADA) and potentially can directly execute the abstract intermediate language output (of the compiler) via microprogramming. This will result in a further improvement in system performance and code execution.

SECTION III

METHOD OF PROCEDURES

INTRODUCTION

The procedure encompassed the analysis of an assumed base-line trainer system. The vehicle chosen for a system design analysis was the F-18 aircraft with the baseline trainer system being limited to an operational flight trainer (OFT). No weapon system for the F-18 was considered for simulation. The performance and life cycle costs of an embedded general purpose computer system were derived and ultimately compared with the cost and performance of a unique multiple microcomputer system design approach capable of performing the same simulation and training functions.

This section provides a description of the baseline Operational Flight Trainer (OFT), a discussion of the projected OFT requirement for the F-18 aircraft, and the methodology employed in determining the processing and storage requirements for such a training system. The results of the analysis determined the performance and number of computers (processors) required to satisfy the F-18 OFT requirements. Two different computer system concepts were investigated. The first represents a conventional approach which utilizes commercially available off-the-shelf general purpose computers. The second is a concept which utilizes a group of microprogrammable microcomputers interconnected and system state controlled to satisfy the OFT simulation requirements.

A survey of large scale integration (LSI) and very large scale integration (VLSI) technology was conducted as part of the overall exploratory development effort. In addition, an analysis was performed to identify various microcomputer modules/families, the advantages and limitations of each, and their potential for being configured to meet the baseline OFT system simulation requirements. The final step in the investigation was to determine a reasonable life cycle cost for each of the two different computer approaches.

TYPICAL OFT SYSTEM CONFIGURATION

An OFT simulates the performance and characteristics during cockpit preflight, engine start operations through taxi and takeoff, navigation, flight, landing and engine shutdown procedures of a specific aircraft. A typical OFT consists of a full size replica of the specific aircraft cockpit mounted on a fixed or motion base. The OFT also includes an instructor station/operator console, a digital computer system with peripheral equipment and a power distribution station all housed in a permanent structure.

System simulation provides for duplication and activation of the controls, performance, position and control instruments, communication and navigation systems and other flight equipment of the cockpit so that trainees can become completely familiar with the cockpit controls and procedures in order to synthesize flight maneuvers and respond to induced emergencies. The trainee activation of the controls and a presentation of instruments/displays duplicates the response of the aircraft throughout its entire operating range.

The instructor station for an OFT is usually located external to the trainee station. It provides trainer control including implementing the flight, inserting malfunctions, monitoring trainee actions and evaluating trainee performance. The instructor's station is utilized to provide information to the trainee concerning the appropriateness of his actions as well as the actions he should have taken at any point during the problem. The instructor station also permits the instructor to maneuver the simulator through a planned flight for demonstration purposes. The instructor's station provides a continuous display of real-time information such as airspeed, altitude, heading, ground position, and the like to the instructor. For simulated aerial navigation and control, data is also available to the trainee in order that he can simulate maneuvering within the flight capabilities of the design basis aircraft. There are also displays which provide information to the instructor pertaining to the operation of the simulator. The displays consist of the following types (1) cockpit instruments, (2) switch position indicators, (3) instrument repeaters, (4) CRT's, and the like.

The embedded computer system for a modern OFT is usually a multiprocessor configuration with private memory, shared or common memory, peripheral equipment, mass storage memory and cockpit input/output data conversion equipment (linkage). This system is typical of the conventional master-slave computer system configuration.[2,3]

One processor is designated as the master unit which controls the trainer system. The master processor usually contains the software for flight, engines, accessories, nav/com, instructor station switches and CRT system, and all training modes. The slave processor handles all other OFT functions.

Figure 1 depicts a basic block diagram of the trainer concept selected for this analysis.

UNIQUE F-18 OFT SIMULATION REQUIREMENTS

Flight Control Computer. The F-18 aircraft utilizes a flight computer system which provides closed loop control of the longitudinal and lateral-direction control. The flight computer system measures pilot pitch, roll and yaw commands, aircraft angular body rates, aircraft linear accelerations and air data parameters. The solution rate of the on-board flight computer system is relatively high, between 40-80 Hz. This imposes a stringent processing

2. Future Avionics Systems Architecture, B.A. Zempolich, Report No. AIR-360, 5 Oct 1977.
3. Microcomputers and Systems Design, W.J. Dejka, TD-507, NELC, 15 Feb 1977.

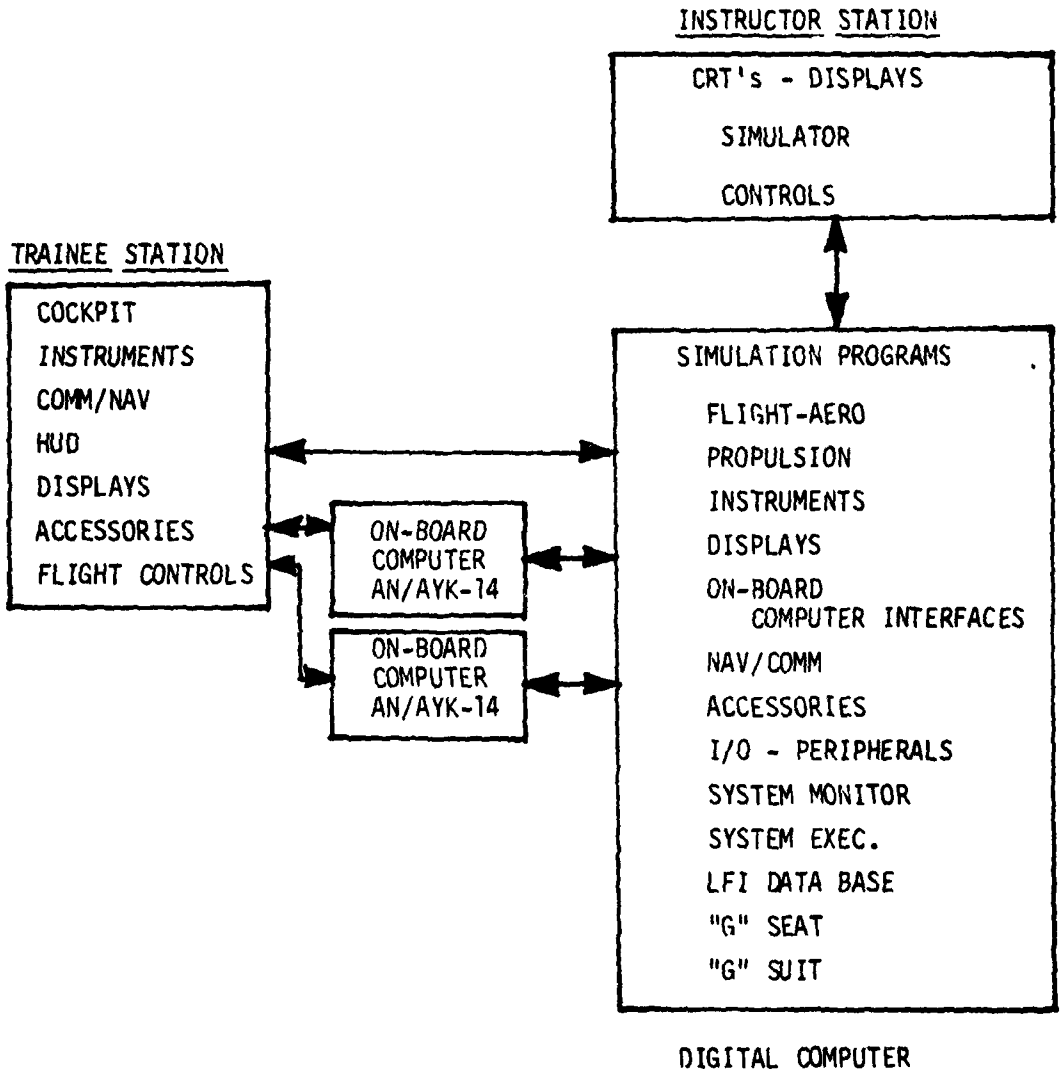


Figure 1. Basic F-18 OFT Block Diagram (Concept)

requirement on the OFT computer system. The OFT math model elements which interface with the on-board flight computer system must be solved at a rate consistent with the performance of that system.

The OFT processing requirement for those functions may be met by either utilizing the on-board flight computer as an integral part of the trainer or by simulating the required functions in the OFT main frame computer system. There are advantages and limitations associated with both approaches. The use of the on-board flight computer system reduces the computational load on the OFT main frame computer system. However, limitations of such an approach are the high cost of the flight computer system, identifying the various interface requirements (hardware and software), reduced system flexibility, and reduced reliability due to system complexity. Simulation of the required on-board flight computer system on the other hand offers a simpler and more reliable system configuration. However, technical design data, both hardware and software, must be available to identify the various on-board control functions and their implementation for simulation. Changes in the simulation software to relect changes in the tactical functions of the on-board computer can be extensive, representing considerable costs in schedule and availability for training to maintain the OFT in an updated status.

Mission Computer. The F-18 aircraft also utilizes two mission computers which provide the primary controls for the avionic system. The mission computers process radar, inertial navigation system (INS) and ADC data for display and weapon delivery computations, utilize store management data to generate missile and weapon release signals, compute navigational and attack steering commands, and control mode of operation of the avionic system as a function of aircraft mode.

In the F-18 the processing requirements associated with the mission computer could be met by either utilizing the operational on-board mission computers or by simulating those functions performed by the mission computer that are required in the OFT. For purposes of this analysis, the mission computers were incorporated in the OFT concept to off load the trainer computer system and facilitate incorporating software/hardware modifications due to future enhancements to the baseline aircraft.

PROCESSING AND STORAGE REQUIREMENTS ANALYSIS

The method of procedure used in determining the storage (memory) and computational requirements for the real-time OFT simulation problem for the F-18 high performance aircraft was that described in a NAVTRAEQUIPCEN report.[4]

4. NAVTRAEQUIPCEN IH-262, Computer System Requirements Analysis, Device 2F112, F14 Weapon System Trainer

A summary of the major computer analysis steps is as follows:

a. Determine all simulation and control functions and designate their program modules.

b. Determine or estimate the number of program statements, data and constant storage requirement for each program module.

c. Determine or estimate the worst-case instruction count for each program module.

d. Determine the computer solution or iteration rate (and frame time) for each program module.

e. Determine the processing requirements for the total OFT simulation program.

f. Derive the percent usage factor for a specified class of typical instructions.

g. Derive an average instruction execution time for the specific computer hardware being considered.

h. Determine computer hardware configurations which will satisfy the performance requirements as determined by e and g above.

i. Determine the computational loading per processor (average number of instructions to be executed per second) which is the sum of the products of the worst-case instruction count/module and the corresponding iteration rate (resulting in a total instructions per second (IPS) figure required of each processor).

j. Perform a frame time analysis to determine if the processors being considered have sufficient processing capability to satisfy the timing requirements on a frame to frame basis.

A significant step in the analysis sequence was to determine the relative capability of computer hardware configurations to process the program involved and meet the program performance requirements.

The instruction repertoires of general purpose digital computers selected for trainer applications have execution speeds which can vary significantly among instruction types. As an example, an ADD-type instruction of one computer may require only 1.5 microseconds for execution and a MULTIPLY-type may require 5.5 microseconds. Whereas, another computer may be capable of executing ADD-type instructions in 1.2 microseconds and MULTIPLY-type in 6.75 microseconds.

In order to normalize such variations from machine to machine and be able to compare two or more for execution capability, the actual computer hardware execution times are weighted by a percent-use factor representing the

frequency each instruction is used in the program. This was done for each computer configuration under consideration.

The percent-use factors (program instruction mix expressed as a percentage) for the conceptual F/A-18 OFT analysis are shown in Table 1. The percent values for this table were derived from an average of actual instruction counts from several existing OFT's and other reasonable estimates. This normalization process provides a relative weighted Average Instruction Execution Time (AIET) for each computer system being considered. Table 2 presents the data for a commercial general purpose computer system that has an AIET per processor of 1.5719 microseconds (equivalent to executing 636,174 instructions per second).

PROGRAM ORGANIZATION AND PERFORMANCE REQUIREMENTS ANALYSIS

The module size, data and constant storage requirements for each functional group were determined by using the methodology described in the NAVTRAEQUIPCEN report[5] as well as actual instruction counts of existing OFT programs and was used in deriving the data of Appendix A.

The estimated storage requirements for each major simulation functional group are summarized in Appendix A. The summation of the instruction count of each identified routine, including appropriate contingency factors, constant and data counts for each module, determines the total storage (memory) requirements for that major functional group. In this analysis, a twenty percent contingency factor was added to each module instruction count to allow for estimating errors due to oversight, changes in training requirements and the like. Since FORTRAN IV was considered to be the programming language for the concepted F-18 OFT, an additional fifteen percent was added to each program module instruction count for all simulation functions to account for inefficiencies in the code generated by FORTRAN compilers as compared to routines coded in assembly level language.

In order to allow for future expansion, spare memory for the F-18 OFT was considered to be not less than twenty-five percent of the installed memory per processor.

WORST-CASE INSTRUCTIONS EXECUTED. In determining the time required to execute the various program modules, it is necessary to determine the worst-case number of instructions of each module which is logically possible to be executed in a single pass through the routine for each computation frame.

The worst-case instruction execution counts for the flight-aero and communication/navigation simulation functional groups for the conceptional F-18 OFT were derived from past experience and other trainer programs with similar processing requirements. The worst-case instruction count for the various flight and communication/navigation modules varied from 0.2 to 6.0 times the module instruction count.

5. See reference 4.

TABLE 1. PERCENT INSTRUCTION MIX/USAGE

INSTRUCTION TYPE	% USAGE
LOAD	0.157
LOAD - DBL. PREC.	0.001
STORE	0.127
STORE - DBL. PREC.	0.001
ADD/SUBT	0.049
ADD/SUTB - FLT PT	0.041
MULTIPLY	0.032
MULTIPLY - FLT PT	0.015
DIVIDE	0.007
DIVIDE - FLT PT	0.001
LOGICAL	0.076
SHIFT (5 PLACES)	0.031
COMPARE	0.043
BRANCH	0.105
INDEX	0.003
REG-TO-REG OPNS	0.031
MISCELLANEOUS (E.G., CALLS TO O/S, ETC)	0.279
INPUT - OUTPUT	0.001
TOTAL	1.000

TABLE 2. COMMERCIAL COMPUTER PERFORMANCE ANALYSIS

INSTRUCTION TYPE	% USAGE	HARDWARE EXECUTION TIME(USEC)	WEIGHTED TIME USED/TYPE (USEC)
LOAD	0.157	1.200	0.1884
LOAD - DBL. PREC.	0.001	2.100	0.0021
STORE	0.127	1.200	0.1524
STORE - DBL. PREC.	0.001	1.800	0.0018
ADD/SUBT	0.049	1.200	0.0588
ADD/SUBB - FLT PT	0.041	2.600	0.1066
MULTIPLY	0.032	5.700	0.1824
MULTIPLY - FLT PT	0.015	5.600	0.0840
DIVIDE	0.007	7.100	0.0497
DIVIDE - FLT PT	0.001	9.050	0.0090
LOGICAL	0.076	1.200	0.0912
SHIFT (5 PLACES)	0.031	1.800	0.0558
COMPARE	0.043	1.950	0.0838
BRANCH	0.105	0.600	0.0630
INDEX	0.003	1.500	0.0045
REG-TO-REG OPNS	0.031	0.600	0.0186
MISCELLANEOUS	0.279	1.500	0.4185
INPUT - OUTPUT	0.001	1.200	0.0012
TOTAL	1.000		1.5719
AVE. INST. EXEC. RATE CAPABILITY (INSTRUCTIONS PER SECOND)			636174

INSTRUCTION EXECUTION RATE. The highest execution rate for the F-18 OFT example was selected to be 60 HZ. The basis for selecting this execution rate was influenced by several factors which are: (1) selected aircraft including interface processing with on-board computers, (2) digital simulation of flight computer functions, (3) future performance considerations, and (4) minimization of overall simulation system time delay and subsystem time lags. Submultiple rates of 60 (i.e., 30, 15, 10, 5, 1) were assigned to other program modules according to the required responses of those modules. Execution time could not exceed total available time in a solution frame (minus twenty-five percent for spare) in order to provide valid system response, simulation performance, and minimization of overall simulation system time delay and subsystem time lag.

The minimum real-time execution rate required by the total simulation program (expressed as instructions per second (IPS)) is the summation of the product of the the worst-case instruction count for each module and required solution rate for that module. Appendix A lists the program modules with assigned solution rates for each of the various simulation functions.

COMPUTER PERFORMANCE ANALYSIS

AVERAGE INSTRUCTION EXECUTION TIME (AIET). The percent usage of each instruction type corresponding to each major group was used to compute the average instruction execution time in microseconds for the computer system being analyzed for the F-18 OFT example.

The F-18 OFT computer analysis indicated that more than a single processor would be required. Consequently, the F-18 OFT simulation task was divided and allotted to three commercial processors (Appendix B). The three-processor system configuration is provided in Appendix C. This system configuration represents a conventional general purpose approach for typical high performance OFT's.

FRAME TIME ANALYSIS. The table of Appendix B and the preceding discussions summarize the minimum program processing requirements for the F-18 OFT example. This table is based on the execution time of all modules being averaged over a full second. This facet of the analysis determined the initial processing and storage capabilities required of the embedded computer hardware.

However, since real-time simulation program execution is basically an iterative sequence of events within the computer, it becomes necessary to analyze the computer requirements to another level of detail to select computer hardware that is fully capable of meeting the processing time requirements per processing frame. This more detailed capability is determined on the basis of a frame time analysis.

For this F-18 OFT example, the highest frame rate for program solution is 60 HZ. This translates to a period of 16.667 milliseconds per frame. Since twenty-five percent spare time per frame under worst-case processing conditions was considered for the analyzed baseline, only 12.500 milliseconds are available in each frame for processing all modules assigned to each frame.

The allocation of the program modules and the results of the frame time analysis for this F-18 example are summarized in Appendix D. It is conceivable that a more optimum (in processing time) allocation of program modules per frame time could be derived than the one selected. However, this should not affect the ultimate hardware configuration. The allocations used in Appendix D indicate that the specified usable processing frame time requirement can be met by the three processor configuration with the F-18 OFT simulation functional assignment indicated in Appendices B and D. This system configuration is a conventional master-slave multiple processor organization.

MULTIPLE MICROCOMPUTER SYSTEM CONCEPTS

Several concepts of system configuration using multiple microcomputers were investigated. These included but were not limited to:

a. A federated approach[6], Figure 2

b. A master-slave approach[8] Figures 3 and 4

c. A matrix approach[8], Figure 5

d. A distributed approach[6,7], Figure 6

FEDERATED APPROACH. The federated approach can provide hardware that can be tailored to various system functions. However, this configuration has a low data rate since the communication bus is treated as a peripheral device. Reconfiguration is difficult to satisfy a wide spectrum of trainer classes. Computers of different architectures could be used to formulate such a system, but this would tend to nullify a standardization requirement for trainers to reduce life cycle costs. From a software viewpoint this configuration would have a single executive or supervisor that would be resident in one of the processors with general system control provided in one unit. Application programs would be limited to only local functions in a single computer. This concept was not investigated further since the low bus data rate would not meet general real-time trainer requirements. With the bus treated as a peripheral serious real-time bus conflicts could develop that will make this technique unacceptable for real-time throughput.

MASTER-SLAVE APPROACH. The master-slave computer system configuration is widely used in sophisticated real-time trainers. Two variations of this approach are shown on Figures 3 and 4. The computer hardware used in these configurations is usually identical except that different models of the same vendor family of computers can be used. Some reconfiguration potential

6. See reference 2
7. See reference 3
8. Parallel Processor Architectures - Part 1, General Purpose Systems, Kenneth J. Thurber, Sperry Univac, Computer Design, January 1979.

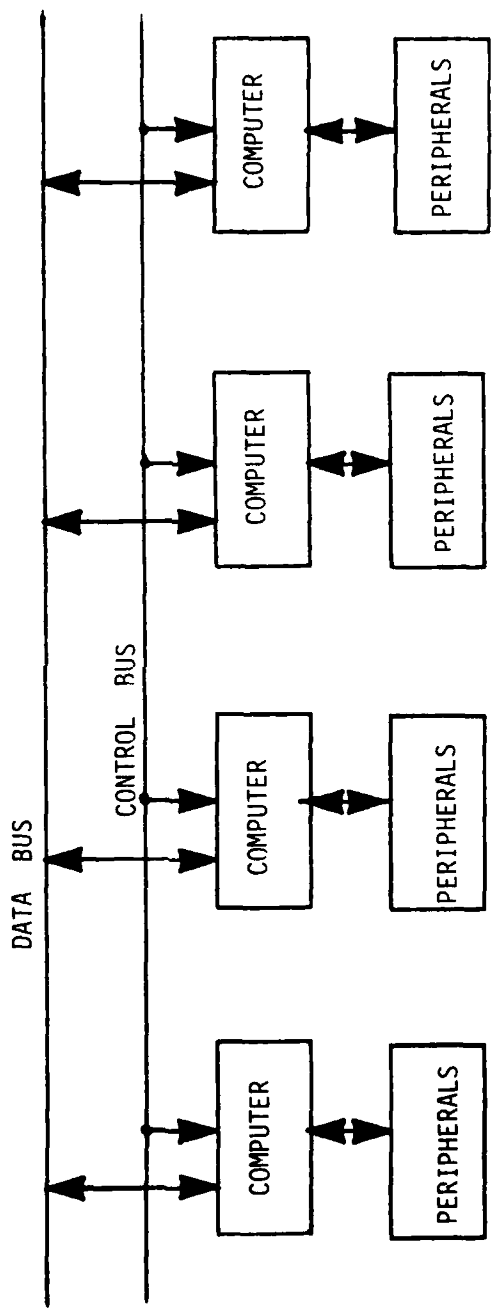


Figure 2. Federated System Concept

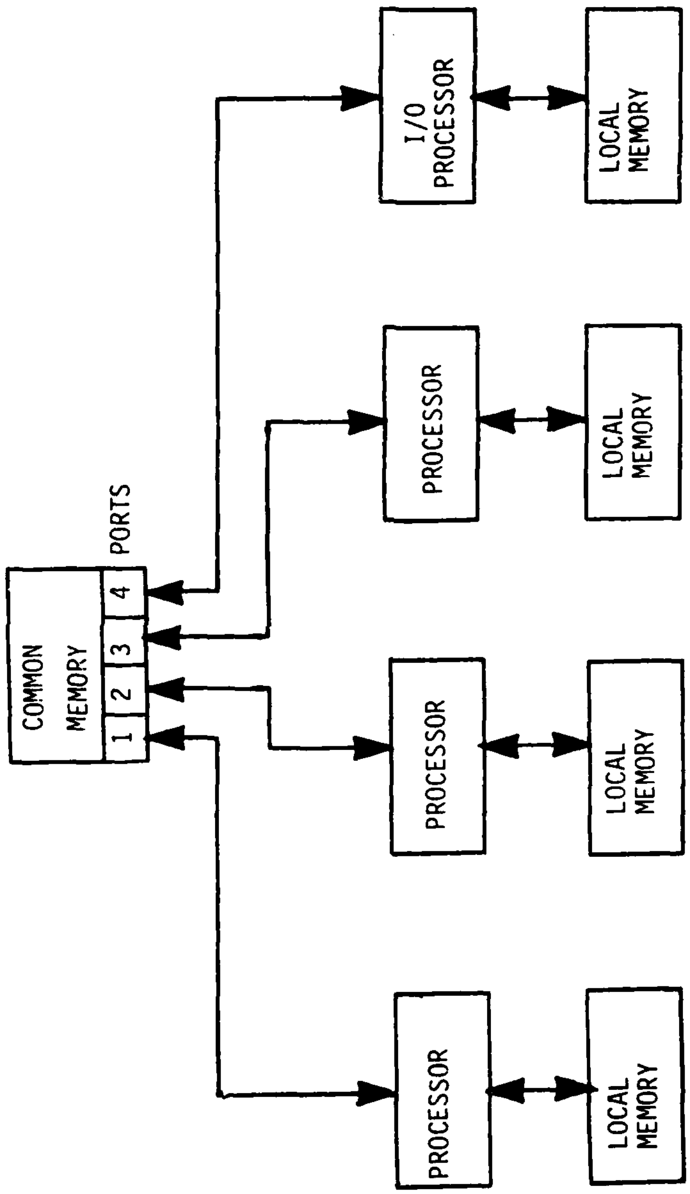


Figure 3. Master-Slave Multiple Processor System Concept - A

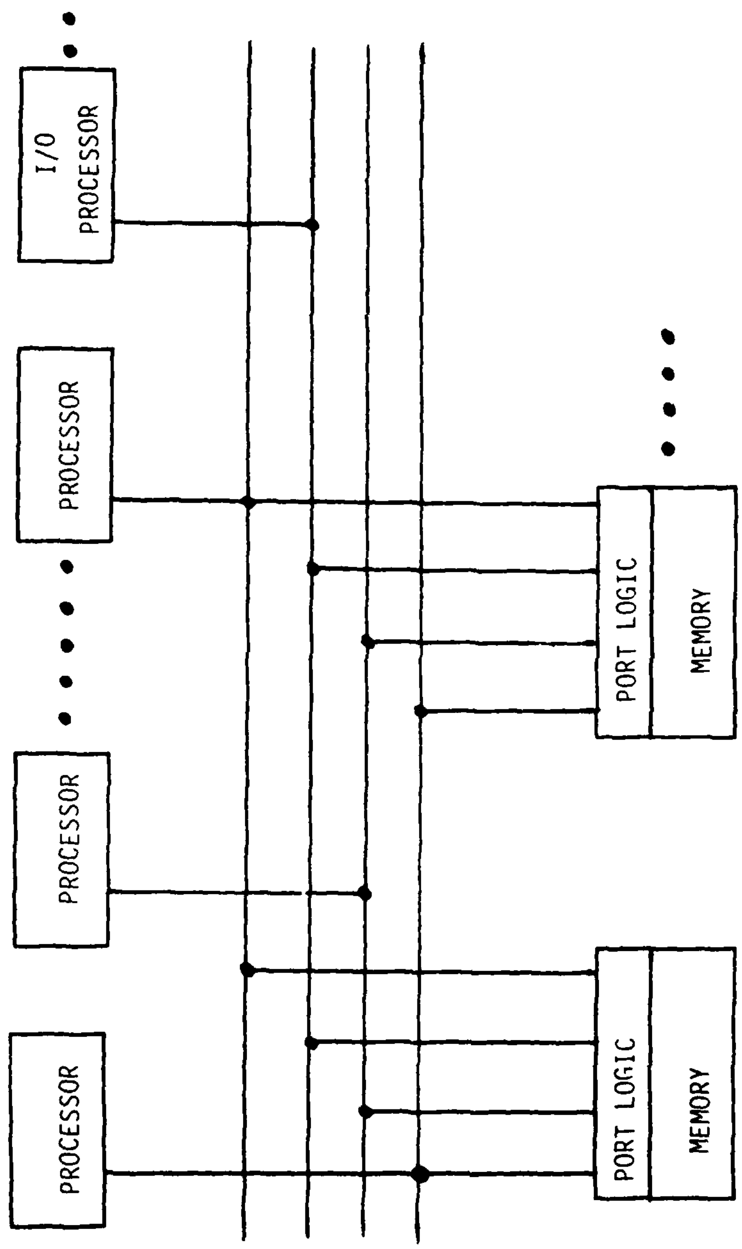


Figure 4. Master-Slave Multiple Processor System Concept - B

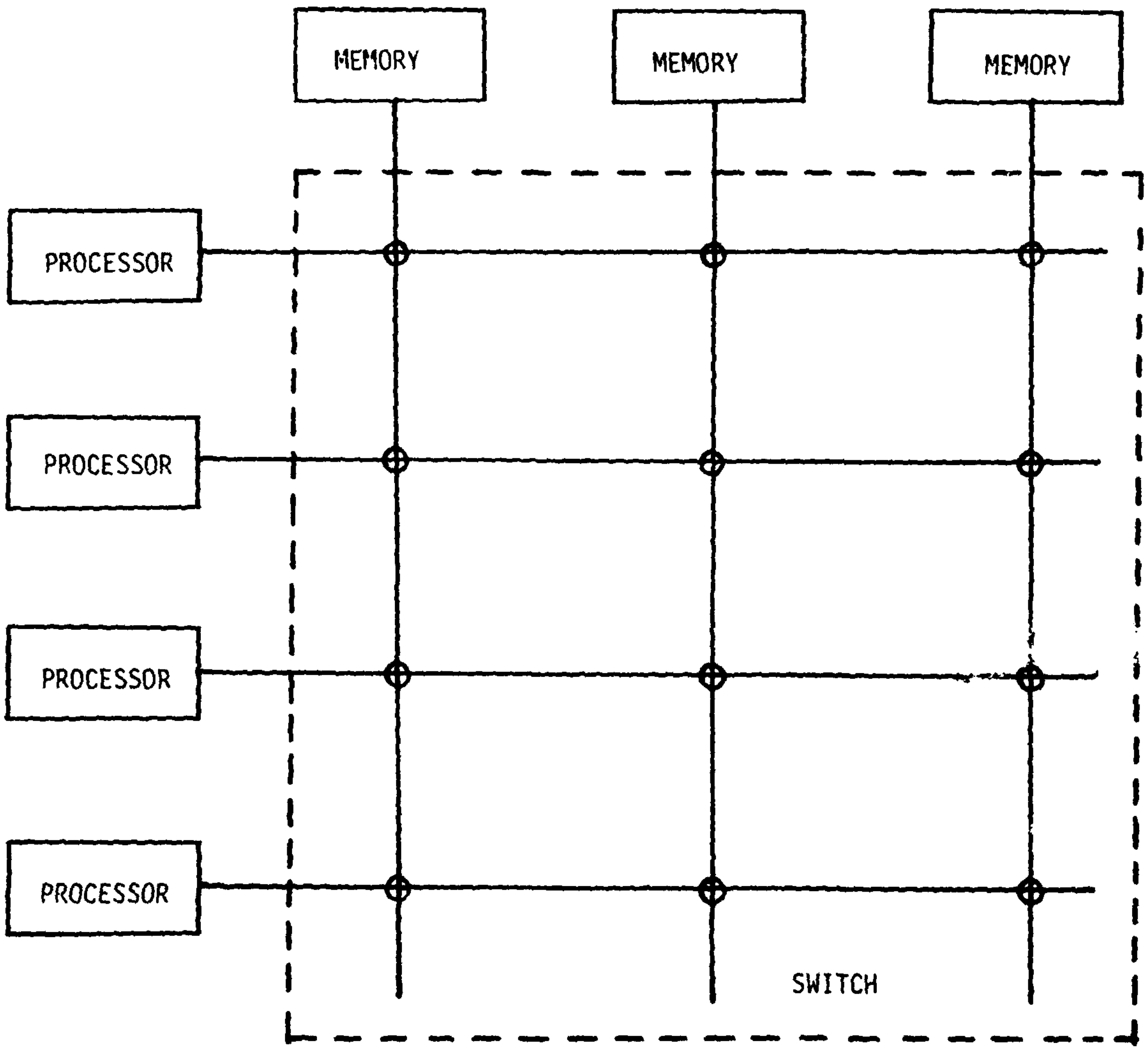


Figure 5. Matrix Computer System Concept

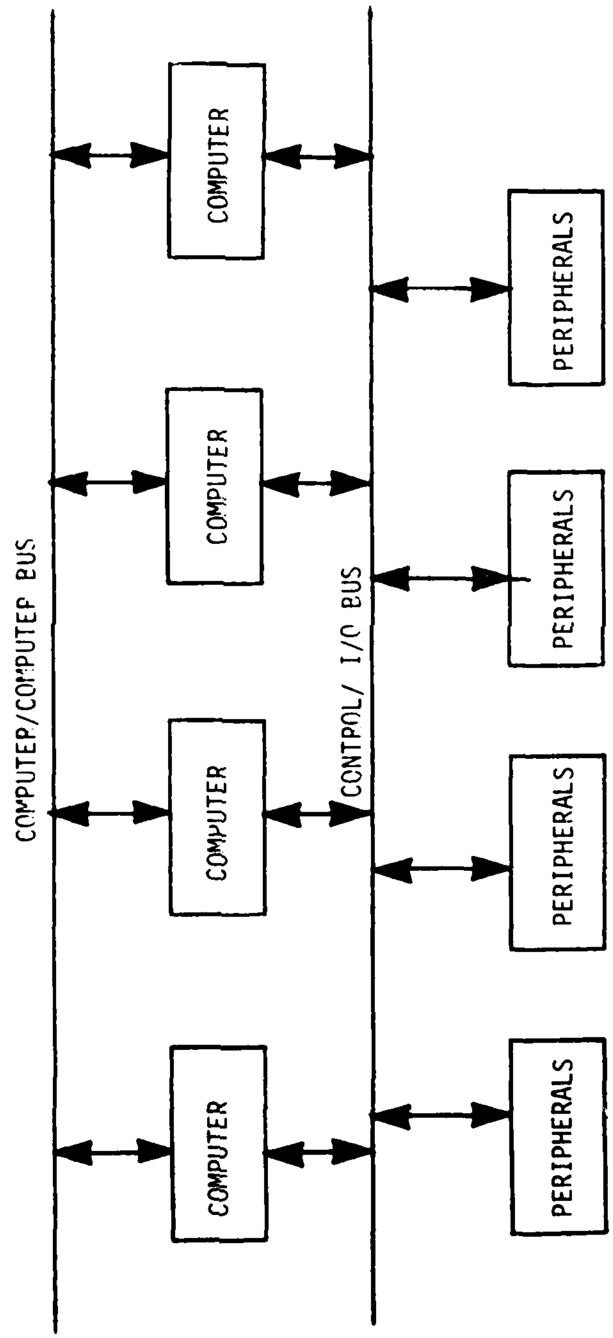


Figure 6. Distributed Computer System Concept

exists, however, it is not usually attempted once the hardware design is firm and proven. Data interchange rates are limited to memory access speeds and the DMA set-up time in the case of the combination of Figure 3 and a common memory is employed. Where multiple memory modules with multiple ports are used (Figure 4), I/O rates are limited by the nature of the programmed functions requiring extensive communications and memory cycle times. Large I/O data rate requirement conflicts will result when the CPU and DMA attempt to access the same address space. There is usually an operating system (or an executive) in the master unit and sub-executives in each slave unit. The system is somewhat tightly coupled (for real-time processing), using either priority interrupts or semaphores between master and slaves. Little or no parallelism is achievable in this configuration when implemented with conventional general purpose computers or microcomputers.

MATRIX APPROACH. The significant advantage of the matrix system configuration is the sharing of processor and memory resources. Figure 5 is a very simplified diagram of a matrix type system configuration. As additional resources are required to handle additional processing requirements the size of the matrix switch increases as the square of the number of resource items. The term "crossbar switch multiple processor" is another name for this type system concept. An access path can be provided to each separate memory module from each processor element. Parallel or simultaneous transfers can occur. System expansion is relatively easy to implement if the switch has been properly designed and implemented. However, the switch is fundamentally complex and difficult to develop. The switch for a system of any reasonable size can represent a substantial part of the total system hardware requirements.

Limitations of this concept for real-time trainer applications can be characterized by memory port design limitations, expensive memory controls and very expensive interconnection cabling required. Memory access conflicts are also extensive. When the number of processors exceeds four or more, significant processing performance degradation occurs. Such limitations make the matrix approach unacceptable for application to real-time trainer designs.

DISTRIBUTED APPROACH. Distributed system approaches are being described in the literature at an ever increasing rate. Distributed processing has become a generic term that must be specifically defined in terms of a specific application. Hardware characteristics usually provide a very high speed bus system that interconnects the several computers in such a network. The network can be either geographically distributed or it can be functionally distributed at a local site (or within several adjacent racks). The ability to reconfigure a system design under this philosophy is dependent upon all computers being identical or all should have the same architecture or must be capable of interfacing to some standard bus. Multiple access to a group of peripherals is also a firm requirement for easy reconfiguration.

Each computer usually has a simple local executive or sub-executive for control. Applications programs are most frequently applicable to local functions in each computer. System control can be distributed via the sub-executive in each computer.

Distributed systems characterized by the above can be implemented to have a graceful degradation quality. This is very important for a tactical system to allow it to continue to perform at least part of a critical mission. Distributed system concepts for real-time trainers will require a much tighter coupling of processing than is the usual distributed case. The concept depicted by Figure 5 is inappropriate for meeting real-time trainer processing requirements.

In each case above, performing the functions of a multiple general purpose computer configuration with multiple microcomputers resulted in lack of flexibility, excessive inter-microcomputer data transfer times, excessive complexity of both hardware and system control software and other undesirable system control requirements (e.g., control busses, data communication bandwidth limitations, etc.). Graceful system degradation objectives of the usual distributed concepts are not applicable to or required by trainer systems.

As a result of investigating the concepts discussed above, another unique system architecture was explored and evolved to develop a concept of using multiple microcomputers that would not exhibit the limitations and unacceptable characteristics of these system concepts. A functionally-modular multiple microcomputer system architectural concept has evolved for performing the general purpose processing, control and computational functions of real-time trainers. The system architectural concept has general purpose characteristics but also possesses inherent properties of excellent flexibility, potential for significant hardware and software standardization, low life cycle cost and improved processing performance. It is uniquely adaptable to a wide spectrum of real-time trainer classes with the capability of being tailored to the processing requirements of a particular trainer. In general, the new era of microelectronics will result in reduced use of conventional general purpose (GP) computer systems in real-time trainers and increased use of application-specific designs using multiple microcomputers.[9]

FUNCTIONALLY-MODULAR MULTIPLE MICROCOMPUTER SYSTEM CONCEPT. This concept is depicted in Figure 7. It involves functionally partitioning the real-time simulation task as conventionally represented in the usual GP stored computer program. The functions required for the CFT example described previously have been partitioned and assigned (or dedicated) to separate but identical microcomputers (Table 3). For example, these functions include control inputs, weight and balance computations, engine simulation, aero-coefficient table look-ups, moment of inertia computations, forces and moments computations, coordinate conversions and transformations, motion system computations, instrument drives, instructor station functions, and the like. For the heuristic partitioning, attempts were made to group related functions in specific microcomputers (Table 3).

The application microcomputers will access a common data memory via a common data bus system and a common memory address bus system. System

9. See reference 2.

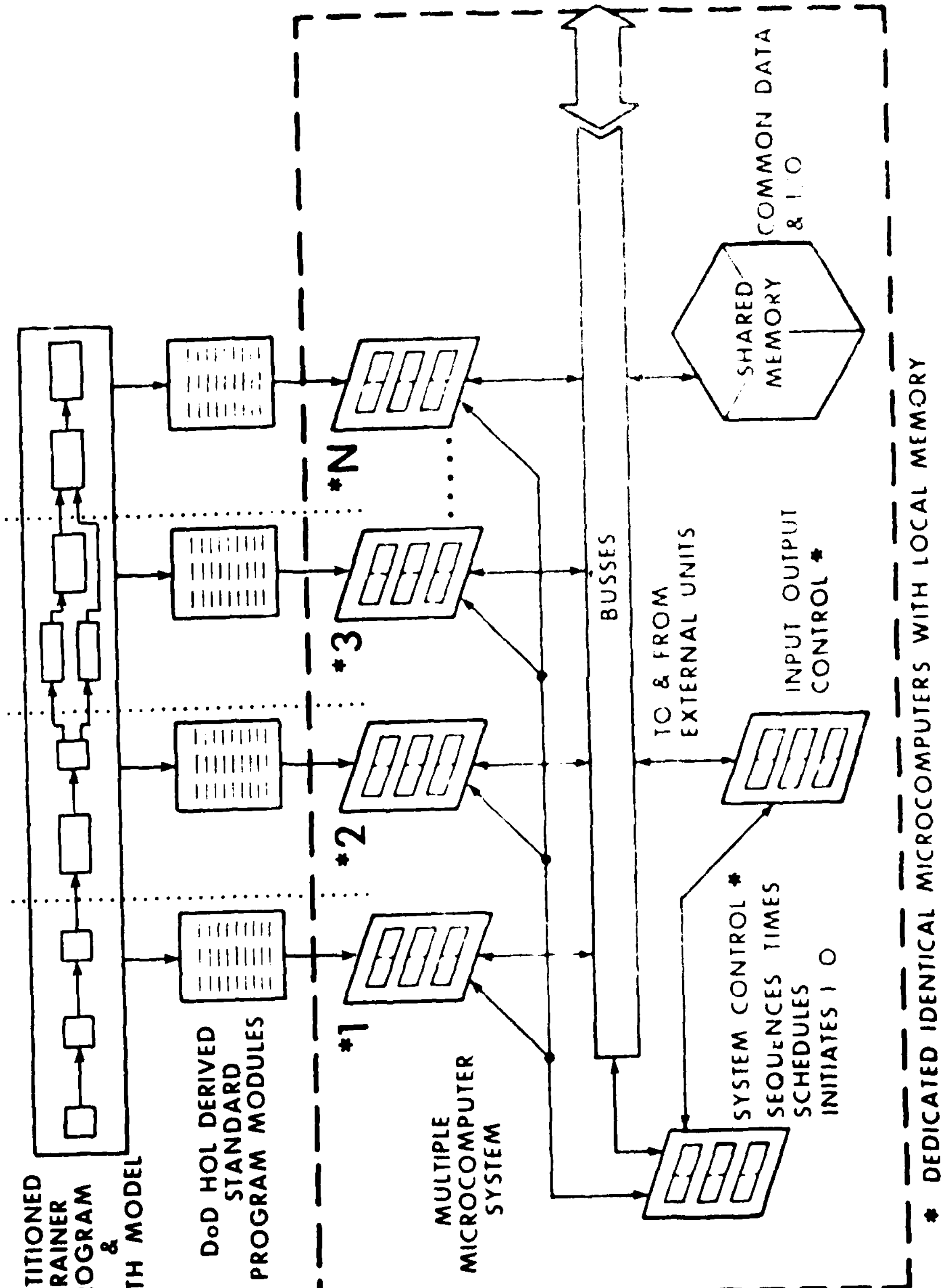


Figure 7. Functionally-Modular Multiple Micro-computer Concept for Trainer Systems

TABLE 3. MICROCOMPUTER DESIGNATIONS (EXAMPLE)

MICROCOMPUTER NO.	SOLUTION RATE	ASSIGNED FUNCTIONS
1	30/sec	Instructor Station, Displays, Ground reaction effects, taxiing
2	60/sec	Aero, Forces, Moments-Stab Axis Comp. Atmosphere, Weight & Balance
3	60/sec	Flight Control, Propulsion system, Flight Control-CPU interface
4	60/sec	Stability Axis transformations, Earth-Axis Acceleration Moments - Body Axis computation P,Q,R Integration, Air Data Comp.
5	60/sec	Angular Positions (Quaternions), G-seat/G-suit, TACAN Math Model
6	30/sec	Navigation, Radar Nav., Compass simulation
7	60/sec	System Control State
8	60/sec	Input-output - Cockpit
9	60/sec	Peripheral units control
10	30/sec	Instructor Station I/O and Controls

control, such as microcomputer frame scheduling, sequencing, bus accessing, frame timing, interrupt status and accounting, common storage access, initiation of input-output operations, etc will be controlled via a dedicated identical microcomputer that has been microcoded and programmed for these system functions. Each application microcomputer will also be microcoded to provide internal local control. This local microcoded control function will be identical in all application and control microcomputers.

The dynamic response of the real-time simulation task that the system control microcomputer establishes, dictates the frame rate and frame timing. The required frame time can be represented by the following expression:

$$T_{FR} = \frac{1}{20 \frac{\omega_n}{2\pi}}$$

where $\frac{\omega_n}{2\pi}$ is the highest natural frequency of the system to be simulated,[10,11]

and $20 \frac{\omega_n}{2\pi} = 10 \frac{\omega_n}{\pi}$

For trainers that have a visual subsystem the following additional condition must be met.

$$\frac{10\,\omega_n}{\pi} \geq 30i \text{ for } i = 1,2,3, \text{ etc (an integer multiplier)}$$

The expression 30i determines that the highest frame or solution rate is not less than 30 HZ or an integer multiple thereof for synchronization with visual systems using the standard TV raster frame rate. For a frame rate of 30 HZ, a Frame will be 33.3333 milliseconds. As an example, for an OFT with visual, the flight equations could be executed at a frame rate of 60 HZ, but the engine equations may require an execution rate of only 10 HZ because of the low natural frequency of the engine.

For this concept various functions are assigned to specific microcomputers to be executed at various rates. Each execution frame that is scheduled by the control microcomputer will involve a group of application microcomputers. Thus, the control microcomputer establishes a total computer system control state for that frame. For the previous example, the multiple microcomputer system would have 60 states.

---

10. Moore School Report 55-20, University of Pennsylvannia, Simulation of a Supersonic Fighter Using a Digital Computer
11. NAVTRADEVCEN 594-1, Analog-Digital Computers For Real-Time Simulation

Frame timing (time length of the frame) will be determined by a precision interval timer (or real-time clock) that is initially set, counts down the frame time, develops an interrupt to the control microcomputer, is reset and resumes counting. These interrupts are counted by the control microcomputer in order to establish the system control state and initiate the applications microcomputers that are required to process during that state. Table 4 illustrates such assignments for various frames. The control state assignments shown on Table 4 are very regular and repetitive. In an actual trainer such assignments and states will not necessarily be regular or very repetitive from state to state (or frame to frame).

Each applications microcomputer communicates data with all other applications microcomputers via a common memory connected to the system data bus. Only simulation initialization data, intermediate results and I/O parameters are stored in common memory.

The control microcomputer will be programmed with a series of pointers and state files (or the like) that list the various groups of applications microcomputers that will be scheduled to process in a given frame. (Table 4 as example). The number of application microcomputers (N) required to process an entire real-time simulation program is determined by the performance capability of the "standard" microcomputer, the bandwidth of the data bus, and the execution time of the total control algorithm in the control microcomputer and in the application microcomputers.

Each microcomputer will have its own dedicated local memory in the form of solid state programmable read-only memory (PROMS) and random access memory (RAMS). Since specific simulation and trainer functions are dedicated to a specific microcomputer, ultimately the software can be standardized on a per-function-basis and stored permanently in a PROM. The RAM will be used only for storing intermediate results and local data and constants. A prime objective of this study and analysis was to determine in a relative way effectiveness of the modular multiple microcomputer system concept. The estimated F-18 OFT program previously described was heuristically partitioned into related processing functions that could be dedicated to individual microcomputers that possessed sufficient processing capability to handle the assigned functions within a specific system control state.

The partitioned functions are shown in Table 3 and in Appendix I. The processing requirements of each functional group were derived in the manner previously explained in this report. Table 5 summarizes the processing requirements of the partitioned functional groups and identifies the number of microcomputers with their minimum average instruction execution time requirement.

It should be noted in Appendix I that the solution rates of all routines assigned to a specific microcomputer have been changed from those of Appendix A to the highest rate required within that group. Otherwise, the scheduling control of separate solution rates within a dedicated microcomputer would be

more complex than needed. This procedure provides for reduced local control firmware complexity and also improves the processing efficiency of a given microcomputer and the total system by eliminating this extra local scheduling overhead.

MICROCOMPUTER LSI AND VLSI TECHNOLOGY INVESTIGATION

The results of the partitioning function explained in the previous section indicated a required AIET performance range of microcomputers for this system concept to be from 2.5991 usec. to 4.7358 usec. This performance requirement range does not include the requirements of the system control microcomputer or the input-output microcomputer(s) (there may be more than one I/O microcomputer).

Eight modular microcompter families were investigated to determine an average instruction execution time capability of each in addition to other essential criteria. Early in the exploratory development described in this report, several decisions were made concerning fundamental criteria to be applied in evaluating families of microcomputer modules.

The essential criteria were:

a. They should be microprogrammable to provide very efficient local control functions as well as to provide a latent capability to potentially capture some existing software, and to achieve hardware commonality between the system control microcomputer, the application microcomputers and the I/O microcomputers.

b. The microprogrammable requirement will permit a future consideration of direct execution of the abstract intermediate level language output from a standard High Order Language (HOL) compiler (e.g., the forthcoming DoD standard HOL, ADA, or PASCAL P-Code).

c. The modular families should be available from multiple commercical sources to preclude sole-source procurements, future field support problems and premature obsolescence.

d. Prime consideration would be given to bit-slice families in order to achieve the microprogrammable capability and word lengths of up to 32 bits with identical modular components.

e. The family of modules should be highly flexible in their application in order that the microcomputers required for a given trainer design could be specifically tailored to the requirements.

f. Instruction execution speeds must be adequate to meet processing times.

TABLE 4. MICROCOMPUTER SYSTEM FRAME/CONTROL STATE SCHEDULING (EXAMPLE)

SYSTEM CONTROL STATE/FRAME NO.	SCHEDULED MICROCOMPUTERS
1	1,2,3,4,5,8,9
2	2,3,4,5,6,8,9,10
3	1,2,3,4,5,8,9
4	2,3,4,5,6,8,9,10
5	1,2,3,4,5,8,9
6	2,3,4,5,6,8,9,10
7	1,2,3,4,5,8,9
8	2,3,4,5,6,8,9,10
"	"
ETC	ETC
"	"
60	2,3,4,5,6,8,9,

TABLE 5. SUMMARY - MEMORY AND AIET REQUIREMENTS

APPLICATION MICROCOMPUTER	ESTIMATED PROGRAM SIZE (WORDS)	ESTIMATED TOTAL MEMORY (WORDS)	REQUIRED AIET (usec)
1	6,784	7,115	4.7358
2	4,725	5,325	2.5991
3	4,556	5,206	3.0157
4	2,717	3,129	4.4215
5	3,459	3,747	2.7821
6	7,298	8,423	3.2949
7	620*	1,500	*
8	100*	750	*
9	100*	750	*
10	1,000*	2,000	*
COMMON MEMORY		16,000	

* Worst-case instruction execution should result in AIET's that are well within the actual AIET of the microcomputer technology analyzed.

The eight different available microcomputer modular families were subjected to a limited performance capability analysis to determine suitability for the concept of Figure 7. A representative average instruction execution time (AIET) was calculated for each. This AIET was derived from a consideration of the microcode required to synthesize the instruction set of Table 1 (less the floating point instructions). The instruction mix (or percent usage) applied in this analysis was that indicated in Appendix H for each family.

Although no precise or exact microcoding was performed, the level to which the instruction synthesis and timing was considered was deemed adequate for this analysis. Figure 8 indicates the modular families considered with a summary qualitative analysis statement for each. A quantitative summary of AIET capability of each microcomputer family analyzed is tabulated in Table 6. Appendix H presents the data used to derive the contents of Table 6. A frame time analysis was performed for the microcomputer approach. The results of this analysis are presented in Appendix J and indicate that the microcomputer family chosen should be capable of performing the necessary OFT processing as described in this report.

TECHNOLOGY REQUIREMENTS

The multiple microcomputer system concept embodied in Figure 7 imposes several other technology requirements for successful achievement. These are listed on Figure 9. Local and global storage (memory) in the form of random access memory elements (RAMS), read-only memory elements (ROMS), programmable read-only memory elements (PROMS), erasable programmable read-only memory elements (EPROMS) and magnetic memory modules are commercially available in many sizes (storage capability) and performance ranges (access/cycle times).

The minimum performance requirements and sizes and types of storage will be determined by a detailed design of the system of Figure 7. Of particular importance is the availability of memory elements with 3-state interface. This capability will permit independent connection to busses regardless of the number of elements (within reason) present or absent and not require a consideration of bus loading. The necessary memory technology is reasonably well established and is available to support this concept.

Static type bussing for interconnection of various numbers of microcomputers can be used in a system design (i.e., require no active components for bus driving or load compensation). It would consist of parallel interconnections, cabling, and static terminations. At least three general types of busses will be required to implement the concept of Figure 7. These are (1) a data transfer bus at least one parallel word in width (32 bits) for transfer of data operands between each microcomputer and the indicated common memory, (2) a control bus for the interchange of system control operands for initiating and controlling the processing frame assignments of the separate microcomputers and the transmission and recognition of bus protocol signals, (3) an address bus for the transmission of microcomputer address information for initiation of specific data transfers to and from all applications microcomputers and the common memory.

MICROCOMPUTER MODULAR FAMILIES INVESTIGATED	ANALYSIS FINDINGS TO DATE
INTEL 3000 BIPOLAR SERIES **	FAST, TOO INFLEXIBLE IN MICROCODE ADDRESSING
TI SPP-9900 SERIES	TOO SLOW, TOO INFLEXIBLE, NOT MICROPROGRAMMABLE, AVAILABLE ONLY FROM A SINGLE SOURCE
FAIRCHILD 9400 BIPOLAR SERIES	TOO SLOW, TOO INFLEXIBLE, NOT MICROPROGRAMMABLE, AVAILABLE ONLY FROM A SINGLE SOURCE
AM 2900 BIPOLAR SERIES **	MICROPROGRAMMABLE, FAST, AVAILABLE FROM AT LEAST 5 COMMERCIAL SOURCES
TI SN74S481 BIPOLAR SERIES **	FAST, MICROPROGRAMMABLE, NOT REGISTER ORIENTED, AVAILABLE ONLY FROM A SINGLE SOURCE
INTEL 8080A MOS MICROPROCESSOR	TOO SLOW, NOT MICROPROGRAMMABLE, TOO INFLEXIBLE
INTEL 8085 A-2 MOS MICROPROCESSOR	TOO SLOW, NOT MICROPROGRAMMABLE, TOO INFLEXIBLE
MOTOROLA MC 10800 MECL SERIE **	FAST, MICROPROGRAMMABLE, AVAILABLE ONLY FROM A SINGLE SOURCE

** BIT SLICE MODULAR FAMILIES

Figure 8. Technology Evaluation Summary

TABLE 6. AVERAGE FIXED POINT INSTRUCTION EXECUTION TIMES

MICROCOMPUTER FAMILY	AIET usec/INST
TI-SPP-9900	14.294
INTEL 3000 BIPOLAR SET	1.926
FAIRCHILD 9440	5.711
INTEL 8080 A	52.782
INTEL 8085 A-2	21.013
AM2900 FAMILY	1.401
MOTOROLA-MC-10800	1.209
TI-SN74S481, 482	1.531

. RAMS, ROMS, PROMS, EPROMS, MAGNETIC CORE	AVAILABLE IN MANY CAPACITIES WITH 3-STATE INTERFACES - 50 NS TO 1000 NS ACCESS TIMES
. BUSSING CONCEPTS	STATIC BUSSES APPEAR TO BE MOST APPROPRIATE - REQUIRE NO ACTIVE COMPONENTS
. PACKAGING CONCEPTS	COMMERCIALLY AVAILABLE IN MANY FORMS & CONFIGURATIONS SUITABLE FOR TRAINER DESIGNS.

Figure 9. Other Required Hardware Technologies

Internal to each microcomputer will also be an address bus, a data bus and a simple control bus for communication to and from local (dedicated) memory by each separate microcomputer.

Packaging concepts and packaging technology have progressed to a point that varieties of proven standard packaging techniques and components are commercially available. It becomes a matter of selecting the type most suited for the system to be assembled.

CRITICAL TECHNOLOGY. The most critical technology required to implement the system architecture concept of Figure 7 is the development of a suitable system control algorithm. This system control algorithm would be implemented by a combination of stored software, stored firmware (microcode) and hardware features of each microcomputer. Combinations of these three elements would be used to synthesize and implement the necessary system control algorithm. An exhaustive search of available professional literature, industry IR&D summaries, NASA and DoD Work Unit Plans, indicates there is no similar system control algorithm available or under development that can support the functionally-modular multiple microcomputer system concept as described in this report. Not only must such a system control algorithm be developed, it must be breadboarded to demonstrate proof of concept feasibility and to evaluate its suitability for the total system control function.

Optimal partitioning of a given simulation problem for assignment to individual microcomputers must also be determined in a more formal manner to optimize the overall processing through-put of the system concept of Figure 7. However, lack of a formal partitioning methodology will not limit the concept since it can be accomplished heuristically in an a priori manner.

CONCEPT FEASIBILITY. Concept feasibility must address both technical and cost issues for full concept acceptance. The conventional general purpose computer approach was described earlier in this report. A life cycle cost model was developed and the GP cost parameters were identified or were derived from vendor price information.

Likewise, a multiple microcomputer concept using the same functions was synthesized and microcomputer cost parameters were applied to the same cost model. The cost model elements and results of cost trade-offs are presented and discussed in the Computer System Ownership Cost Model Section that follows.

COMPUTER SYSTEM OWNERSHIP COST MODEL

A life cycle cost model was developed to identify the major cost elements which contribute to the life cycle cost of a trainer computer system. The useful life of a trainer and therefore the computer system was estimated to be ten years. The life cycle cost model (Appendix E) has three major cost factors. These are (1) initial hardware cost, (2) computer system acquisition cost, and (3) ten year support cost of the computer system.

Two different computational system approaches were investigated, analyzed, and configured to meet the OFT processing and storage requirements identified in the earlier discussions. One computational system configuration represents a conventional approach. This system is comprised of three off-the-shelf general purpose computers and associated peripheral equipment that operate in a master-slave configuration. A commercially available computer with hardware floating point was chosen as an analysis baseline since detailed performance data and GSA cost schedules were available. This approach and system configuration concept is depicted in Appendix D. The other computational system approach utilizes ten microcomputers and associated peripherals. This system approach is based on current state-of-the-art microcomputer technology and represents a unique approach in satisfying the stated OFT computational requirements. This approach and system concept was described in a previous section of this report.

The computer system configuration for both approaches was derived by matching the performance of each computational system with the processing and storage requirements summarized in Appendix A. The results of the computer system performance analysis indicates that each system configuration is capable of meeting the baseline OFT computational requirements.

The next step in the analysis was to determine which approach is the most economical system to own and operate for a period of ten years. This was accomplished by utilizing the life cycle cost model identified in Appendix E for each approach. This model provides a realistic basis as well as a systematic approach for identifying the various cost factors which contribute to the total life cycle cost of a typical trainer computer system. Details of the ownership cost analysis for each of the two approaches are provided in Appendices F and G for the conventional system, and Appendices K and L for the multiple microcomputer system. A comparative cost summary of the two for each of the major ownership cost elements is provided below and in Appendix M.

a. Hardware - as of the time of this report preparation (1979), the hardware cost for the conventional computational system for this OFT example is \$447 K compared with \$228 K for the microcomputer system. The microcomputer approach represents a total hardware cost saving of approximately \$219 K or forty-nine percent when compared to the conventional GP approach.

b. Acquisition - this heading consists of the following cost factors (1) simulation software development effort, (2) initial computer system spares, (3) maintenance training and (4) test equipment. The cost for the microcomputer approach represents a savings of approximately \$192 K or forty percent. There are several reasons for this. The simulation software effort for the microcomputer system concept is less extensive since a portion of the overall simulation software program can ultimately be standardized. In addition, this system approach does not utilize an operating system as does the conventional GP approach that requires software modifications to satisfy the real-time simulation task requirements.

The cost of provisioning spare parts for the microcomputer approach is less than the GP system since each microcomputer consists of identical and cheaper hardware modular elements. The attribute which makes each microcomputer system unique for different applications is the software program and control firmware that resides in PROM and ROM for each microcomputer.

The maintenance training cost of this approach is also less since each microcomputer performs a dedicated function. Maintenance personnel troubleshoot the computer system to the functional level and replace the corresponding hardware module(s). This reduces the need for complex and expensive test equipment and reduces the overall system down time.

No attempt has been made to identify potential cost saving attributed to greater system availability due to reduced mean-time to repair. In training systems which operate on a two or three shift schedule, the savings could be substantial.

c. Ten year support - the following cost factors were included under this heading: (1) computer hardware spares, (2) hardware maintenance personnel and (3) software maintenance personnel. The microcomputer system approach is the less costly of the two systems to maintain and support over a ten year period and represents a savings of approximately $420 K or fifty-two percent. This is due to a high degree of hardware module commonality possible and the reduced level of hardware and software effort required to support this system concept.

SECTION IV

RESULTS AND CONCLUSIONS

The computer system requirements for the baseline OFT were anlyzed by a method that is generally accepted for performance determination. The methodology has been described or referenced and the derived data is provided in the appendices. The software requirements for the baseline F-18 example indicates the computer processing capability must be at least 1.24 million operations per second. This equates to an average instruction execution time (AIET) requirement of 0.8071 microseconds.

An actual OFT instruction-use mix was used to derive an AIET capability of a commercially available computer. Comparing the AIET required by the real-time program with the AIET of the baseline computer equipment indicates the OFT computer system must consist of at least three commercial computers to satisfy all real-time computation, simulation and control requirements.

A three computer system was configured and a life cycle cost figure derived from the cost model previously explained.

The same program was partitioned in an heuristic manner by grouping related processing functions. The processing requirements of each of these groups was derived and compared with the capability of the several microcomputers previously analyzed. Since a microcomputer comprised of the AM 2900 series bit-slice modular family provided the necessary capability, the component cost of the required number of microcomputers was determined from vendor supplied application data and price lists.

A total of ten microcomputers was required to meet the program performance requirements for the OFT as partitioned in the manner indicated. Six microcomputers were dedicated to handle the total simulation and processing, three were required to control all input-output functions and one was dedicated as the system processing state controller.

The life cycle cost of the multiple microcomputer approach was derived using the same cost model as for the commercial computer approach.

A relative overall cost comparison for a ten year life cycle indicates a basic saving of approximately $831 K (forty-eight percent) between the general purpose computer approach using multiple commercial computers and the functionally-modular multiple microcomputer approach. This relative saving figure can change when the concept is applied to different classes of trainers. The less complex the trainer, the less the cost should be.

Technologies required for successful achievement of the multiple microcomputer concept have also been discussed considering availability and technical risk. All major hardware technologies are commercially available. The required technology with the technical risk is the system control algorithm that the dedicated system control microcomputer and applications microcomputers must execute. An exhaustive search and review of current industry IR&D programs, NASA R&D and DoD R&D Work Unit Plans indicates there is no R&D effort either in being or planned that will provide the detailed control capability required for implementing the concept for high performance real-time trainers. The development of a suitable control algorithm is therefore necessary.

It can be concluded that a functionally modular microcomputer system architecture as described in this report and depicted in Figure 7 is conceptually feasible with state-of-the-art technology and will be cost effective.

Project 5741, Phase I exploratory development, has been initiated for research and development of the required hardware-firmware-software control algorithm. A concept feasibility breadboard will also be required (and is planned as Phase II) to demonstrate and evaluate feasibility of the concept.

REFERENCES

1. Bus Interface Unit Design for the Distributed Processor/Memory System - Thesis - AFIT/GE-EE/77-24, Leslie T. Konno - AD-A053269 - (functional decentralization and distributed system concept)

2. A comparison of Tactical Military and Commercial Data Processing Computer Architectural Requirements - NUSC Tech Report 5783 - 23 May 1978, Lynn A. DeNois, Robert L. Gordon - AD A055975 Comparison of Architectural Requirements

3. A 16-Bit Microcomputer for Missile Guidance and Control Applications Dr. Eugene H. Lowe, Martin Marietta Corp., Orlando, FL - Joint Automatic Control Conference, San Francisco, CA, 1977, AIAA Proceedings

4. "Navy Distributed Computer Systems Workshop", Naval Surface Weapons Center, Dahlgren Laboratory (K70) Dahlgren, VA 22448, April 1978, Discussed the impact, payoff and technical issues of distributed computer systems for tactical Navy implementaiton.

5. "Emulation of Computer Networks by Microprogrammable Microcomputers" The Computer and Information Science Research Center, The Ohio State University, Report No. OSU-CISRC-TR-74-4, June 1974

6. Performance & Control of Multiple Microprocess Sys., Computer Design Vol 13 3 (March 1974), pp 81-86

7. "Computer Modules: An Architecture for Large Digital Modules", Proc. First Annual IEEE Symp on Comp Arch, Dec 1973, pp 231-237

8. "Parallel Processor Architectures - Part 1, General Purpose Systems" Kenneth J. Thurber, Sperry Univac, Computer Design, Vol 18 No 1 (January 1979), pp 89-97

9. "Parallel Processor Architectures - Part 2 Special Purpose Systems" Kenneth J. Thurber, Sperry Univac, Computer Design, Vol 18 No 2 (Feb 1979) pp 103-114

10. "Dynamic Problem Oriented Redefinition of Computer Architecture Via Microprogramming" PhD Dissertation, T. G. Rauscher, University of Maryland, 1975

11. Microcomputers and Systems Design, William J. Dejka, 15 Feb 1977, Technical Document 507, Naval Electronics Laboratory Center

12. Future Avionics Systems Architecture, B. A. Zemploich, NAVAIRSYSCOM-AIR 360, Report No. AIR 360:BAZ - 5 Oct 1977

13. Microprocessor Arrays for Large Scale Computation, William H. Karitz SRI International, Menlo Park, CA, N78-19816

14. Computing Technology in the 1980's, Harold S. Stone, Visiting Professor, University of California, Berkeley, CA N78-19780

15. Microprogamming Handbook - Second Edition - Advance Micro Devices, Inc., 1977

16. Computer (IEEE Computer Society), February 1979, (1978 Asilomar Workshop on Microprocessors)

17. An Introduction to Microcomputers, Volume II - Some Real Products - June 1977 Revision, Osborne and Associates, Inc., Berkeley, CA

18. Microprocessor Systems Design, Edwin E. Klingman, Prentice Hall, Inc., 1977

19. Microprocessors - Technology, Architecture and Applications, Daniel R. McGlynn - John Wiley & Sons - 1976

20. Microprogramming - Principles and Practices - Samir S. Husson, Prentice Hall -1970

21. The Efficiency of FORTRAN in Simulation Computers, F. A. Sigmund, Goodyear Aerospace Corp., Proceedings, 10th NTEC/Industry Conference, NAVTRAEQUIPCEN IH-294, Nov 1977

22. NAVTRAEQUIPCEN IH-262, Computer System Requirements Analysis, Device 2F112, F14 Weapon System Trainer

APPENDIX A

BASELINE F-18 PROGRAM PROCESSING AND STORAGE REQUIREMENTS

In order to minimize the number of tables/pages and to facilitate the readability and understanding of the various tables, many of the program modules have been grouped together based upon their solution rates and the CPU to which they have been assigned. The modules, their solution rates, and the various program routines which comprise a given module are identified below.

Module 1 (60 hz) This module consists of the Flight Control System, Flight Control/CPU Interface, Aero, Forces and Moments.

Module 2 (60 hz) This module consists of Transformations, aircraft velocities and Accelerations.

Module 3 (60 hz) This module consists of Moments-Body Axis and P,Q,R Dot Integration.

Module 4 (30 hz) This module consists of (A) aircraft angular position, weight and balance, atmospheric simulation, air data computation, TACAN model; and (B) G-Seat and G-Suit.

Module 5 (15 hz) This module consists of (A) On-Board Computer Interface; (B) Strut Deflection Rates, Vertical Strut Forces, Radar Navigation Facilities Simulation and cockpit Instruments; (C) Longitudinal, Lateral Gear Forces, Total Gear Forces, Total Gear Moments, Compass Simulation and Cockpit Instruments (D) Navigation Data Computation, UHF-ADF Simulation, and Radar Altimeter Simulation.

Module 6 (10 hz) This module consists of (A) Propulsion System; (B) Nose Wheel Steering; (C) Ground Turning Rates; (D) Instructor Station; (E) Instructor Displays.

Module 7 (5 hz) This module consists of(A) Brake Forces and (B) Accessary Simulation.

Module 8 (60 hz) Executive and Real-Time Operating System

Module 9 (60 hz) Sub-Executive CPU 1

Module 10 (60 hz) Sub-Executive CPU 2

TABLE A-1. PROGRAM PROCESSING AND STORAGE REQUIREMENTS

PROGRAM FUNCTION	INSTRUCTION ESTIMATES	CONSTANT & DATA ESTIMATES	TOTAL MEMORY REQUIRED	WORST-CASE INSTRUCTION PER SOL.CYC.	WORST-CASE INSTR. EXEC. RATE
MODULE 1 (60 HZ)	4,455	580	5,035	5,872.5	352,350
MODULE 2 (60 HZ)	1,190	215	1,405	1,692.5	101,555
MODULE 3 (60 HZ)	311	85	396	580.5	34,830
MODULE 4 (30 HZ)	4,388	530	4,918	6,682.5	200,475
MODULE 5 (15 HZ)	5,617	900	6,517	7,847.2	117,907
MODULE 6 (10 HZ)	5,670	310	5,980	6,203.3	62,033
MODULE 7 (5 HZ)	2,005	105	2,110	2,205.2	11,027
MODULE 8 (60 HZ)	19,575	250	19,825	1,849.5	110,970
TOTALS WO/SPARE	43,211	2,975	46,186	-----	991,147
SPARE (25%)	10,803	744	11,574	-----	247,787
TOTALS W/SPARE	54,014	3,719	57,733	-----	1,238,943

AVERAGE INSTRUCTION EXECUTION TIME.......................... 0.8071 MICROSECONDS

APPENDIX B

PROCESSING AND STORAGE REQUIREMENTS PER PROCESSOR BASED ON F18 OFT REQUIREMENTS

TABLE B-1. PROGRAM PROCESSING AND STORAGE REQUIREMENTS

PROGRAM FUNCTION CPU # 1	INSTRUCTION ESTIMATES	CONSTANT & DATA ESTIMATES	TOTAL MEMORY REQUIRED	WORST-CASE INSTRUCTION PER SOL.CYC.	WORST-CASE INSTR. EXEC. RATE
MODULE 1 (60 HZ)	4,455	580	5,035	5,872	352,350
MODULE 9 (60 HZ)	810	50	860	810	48,600
TOTALS WO/SPARE	5,265	630	5,895	------	400,950
SPARE	1,316	158	1,474	____	100,238
TOTALS W/SPARE	6,581	788	7,369	____	501,188

AVERAGE INSTRUCTION EXECUTION TIME.......................... 1.9953 MICROSECONDS

TABLE B-2. PROGRAM PROCESSING AND STORAGE REQUIREMENTS

PROGRAM FUNCTION CPU # 2	INSTRUCTION ESTIMATES	CONSTANT & DATA ESTIMATES	TOTAL MEMORY REQUIRED	WORST-CASE INSTRUCTION PER SOL.CYC.	WORST-CASE INSTR. EXEC. RATE
MODULE 1 (60 HZ)	1,190	215	1,405	1,683	101,555
MODULE 4 (30 HZ)	4,402	540	4,942	6,696	200,880
MODULE 10 (60 HZ)	810	50	860	810	48,600
TOTALS WO/SPARE	6,402	805	7,207	9,189	351,035
SPARE	1,600	201	1,802	----	87,759
TOTALS W/SPARE	8,002	1,006	9,009	----	438,794

AVERAGE INSTRUCTION EXECUTION TIME.......................... 2.279 MICROSECONDS

TABLE B-3. PROGRAM PROCESSING AND STORAGE REQUIREMENTS

PROGRAM FUNCTION CPU # 3	INSTRUCTION ESTIMATES	CONSTANT & DATA ESTIMATES	TOTAL MEMORY REQUIRED	WORST-CASE INSTRUCTION PER SOL.CYC.	WORST-CASE INSTR. EXEC. RATE
MODULE 3 (60 HZ)	311	85	396	580	34,830
MODULE 5 (15 HZ)	5,603	890	6,493	7,833	11,7502
MODULE 6 (10 HZ)	5,670	310	5,980	6,203	62,033
MODULE 7 (5 HZ)	2,005	105	2,110	2,205	11,027
MODULE 8 (60 HZ)	19,575	250	1,9825	1,849	110,970
TOTALS WO/SPARE	33,164	1,640	34,794	______	336,362
SPARE	829	410	8,699	______	84,091
TOTAL W/SPARE	41,455	2,050	43,493	-------	420,453

AVERAGE INSTRUCTION EXECUTION TIME.......................... 2.3784 MICROSECONDS

APPENDIX C

FRAME TIME ANALYSIS - CONVENTIONAL APPROACH

TABLE C-1. EXECUTION TIME PER FRAME - MILLISECONDS

PROGRAM FUNCTION CPU # 1	WORST-CASE INSTRUCTION EXECUTED	FRAME 1 THROUGH 60
MODULE 1 (60 HZ)	5873	9.232
MODULE 9 (60 HZ)	810	1.273
TOTAL TIME PER FRAME BASED UPON 1.572 u sec AIET.		10.505

TABLE C-2. EXECUTION TIME PER FRAME - MILLISECOND

PROGRAM FUNCTION CPU # 2	WORST-CASE INSTRUCTION EXECUTED	FRAME ODD 1 THROUGH 59	FRAME EVEN 2 THROUGH 60
MODULE 3 (60 HZ)	1694	2.662	2.662
MODULE 4A (30 HZ)	3376	-----	5.307
MODULE 4B (30 HZ)	3308	5.200	-----
MODULE 10 (60 HZ)	810	1.273	1.273
TOTAL TIME PER FRAME BASED UPON 1.572 u sec AIET.		9.135	9.242

TABLE C-3. EXECUTION TIME PER

CPU #3		FRAME	FRAME	FRAME	FRAME	FRAME	FRAME	FRAME	FRAME	FRAME
PROGRAM FUNCTION	WCIE*	1,13, 25,37, 49	2,14, 26,38, 50	3,15, 27,39, 51	4,16, 28,40, 52	5,17, 29,41, 53	6,18, 30,42, 54	7,19, 31,43, 55	8,20, 32,44, 56	9,21, 33,45, 57
MODULE 3 (60 HZ)	338	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531
MODULE 5A (15 HZ)	2700	4.244	-----	-----	-----	4.244	-----	-----	-----	4.244
MODULE 5B (15 HZ)	1694	-----	2.663	-----	-----	-----	2.663	-----	-----	-----
MODULE 5C (15 HZ)	1765	-----	-----	2.774	-----	-----	-----	2.774	-----	-----
MODULE 5D (15 HZ)	1674	-----	-----	-----	2.632	-----	-----	-----	2.632	-----
MODULE 6A (10 HZ)	2531	-----	3.979	-----	-----	-----	-----	-----	3.979	-----
MODULE 6B (10 HZ)	149	-----	-----	0.234	-----	-----	-----	-----	-----	0.234
MODULE 6C (10 HZ)	149	-----	-----	0.234	-----	-----	-----	-----	-----	0.234
MODULE 6D (10 HZ)	675	-----	-----	1.061	-----	-----	-----	-----	-----	1.061
MODULE 6E (10 HZ)	2700	-----	-----	-----	4.244	-----	-----	-----	-----	-----
MODULE 7A (5 HZ)	1931	-----	-----	3.036	-----	-----	-----	-----	-----	-----
MODULE 7B (5 HZ)	275	0.432	-----	-----	-----	-----	-----	-----	-----	-----
MODULE 8 (60 HZ)	1850	2.908	2.908	2.908	2.908	2.908	2.908	2.908	2.908	2.908
TOTAL TIME PER FRAME BASED UPON 1.572 u sec. AIET.		8.115	10.081	10.778	10.315	7.683	6.102	6.213	10.050	9.212

* ESTIMATED WORST-CASE INSTRUCTIONS EXECUTED

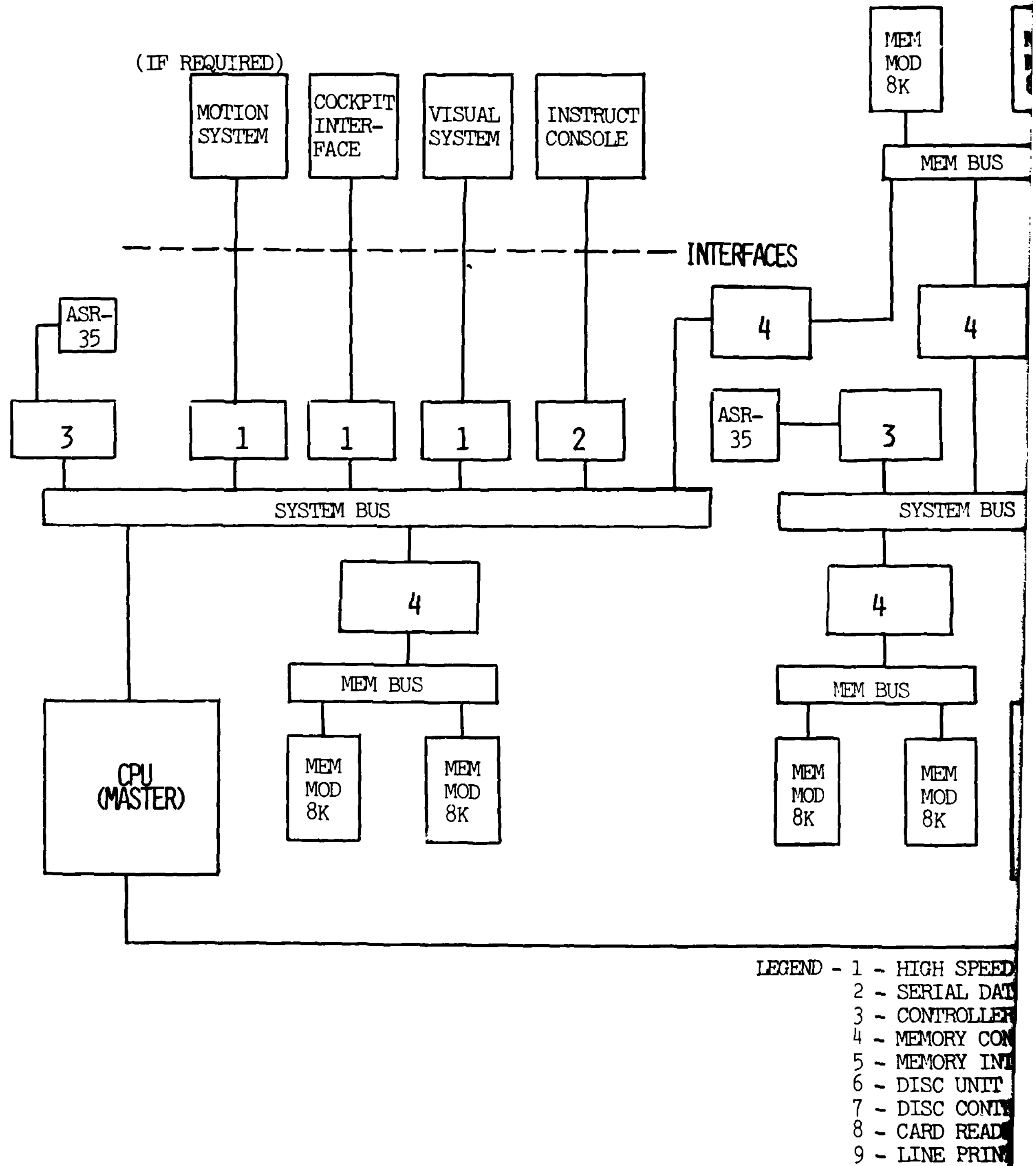
TABLE C-3. EXECUTION TIME PER FRAME - MILLISECONDS

	FRAME	FRAME	FRAME	FRAME	FRAME	FRAME	FRAME	FRAME	FRAME	FRAME	FRAME	FRAME
IE*	1,13, 25,37, 49	2,14, 26,38, 50	3,15, 27,39, 51	4,16, 28,40, 52	5,17, 29,41, 53	6,18, 30,42, 54	7,19, 31,43, 55	8,20, 32,44, 56	9,21, 33,45, 57	10,22, 34,46, 58	11,23, 35,47, 59	12,24, 36,48, 60
338	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531	0.531
700	4.244	-----	-----	-----	4.244	-----	-----	-----	4.244	-----	-----	-----
694	-----	2.663	-----	-----	-----	2.663	-----	-----	-----	2.663	-----	-----
765	-----	-----	2.774	-----	-----	-----	2.774	-----	-----	-----	2.774	-----
674	-----	-----	-----	2.632	-----	-----	-----	2.632	-----	-----	-----	2.632
531	-----	3.979	-----	-----	-----	-----	-----	3.979	-----	-----	-----	-----
149	-----	-----	0.234	-----	-----	-----	-----	-----	0.234	-----	-----	-----
149	-----	-----	0.234	-----	-----	-----	-----	-----	0.234	-----	-----	-----
675	-----	-----	1.061	-----	-----	-----	-----	-----	1.061	-----	-----	-----
700	-----	-----	-----	4.244	-----	-----	-----	-----	-----	4.244	-----	-----
931	-----	-----	3.036	-----	-----	-----	-----	-----	-----	-----	-----	-----
275	0.432	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
850	2.908	2.908	2.908	2.908	2.908	2.908	2.908	2.908	2.908	2.908	2.908	2.908
E.	8.115	10.081	10.778	10.315	7.683	6.102	6.213	10.050	9.212	10.346	6.213	6.071

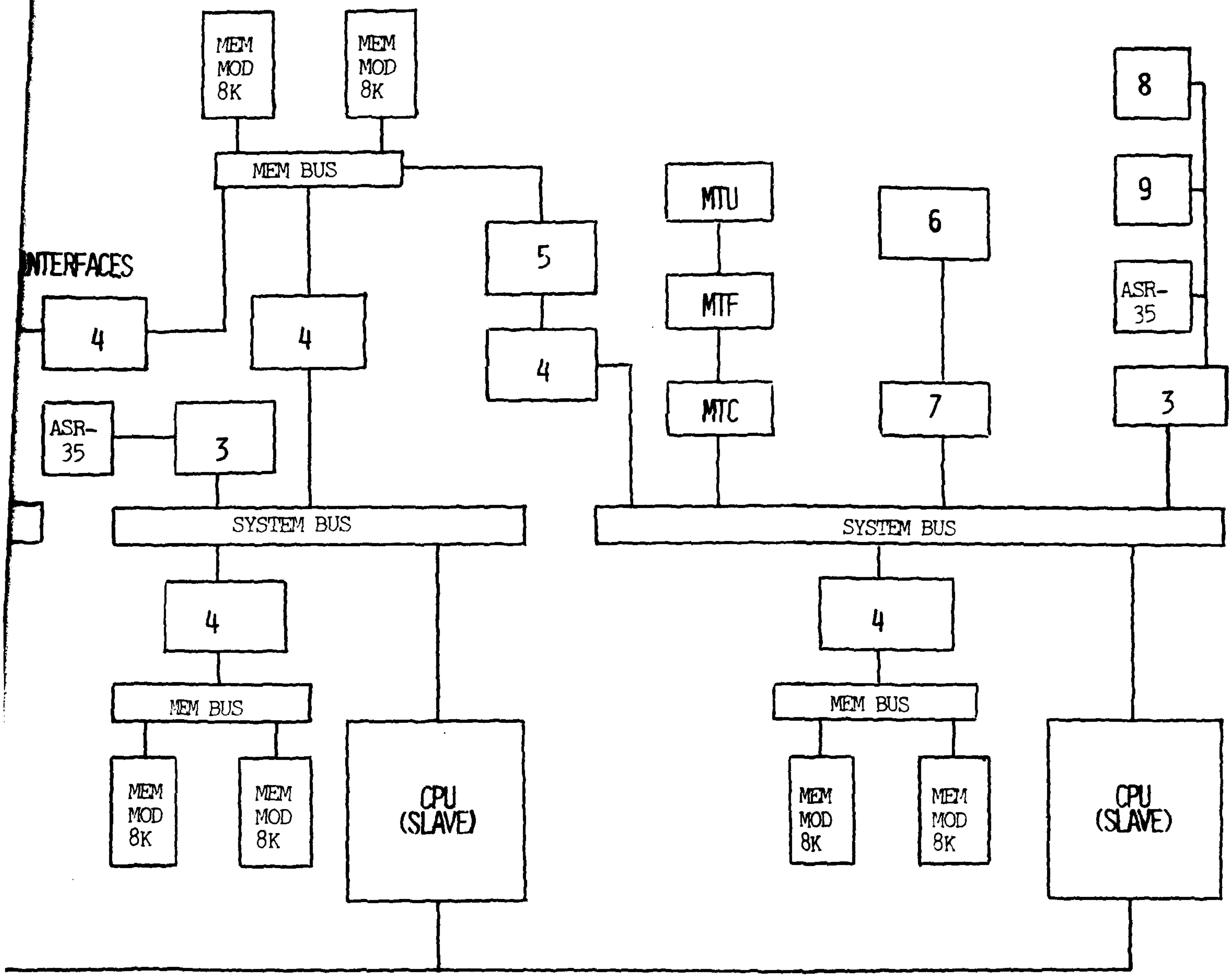
E INSTRUCTIONS EXECUTED

APPENDIX D

CONVENTIONAL COMPUTER SYSTEM CONFIGURATION



(IF REQUIRED)
MOTION SYSTEM
COCKPIT INTER-FACE
VISUAL SYSTEM
INSTRUCT CONSOLE
INTERFACES
MEM MOD 8K
MEM BUS
ASR-35
4
3
1
1
1
2
4
ASR-35
3
SYSTEM BUS
SYSTEM BUS
4
4
MEM BUS
MEM BUS
CPU (MASTER)
MEM MOD 8K
MEM MOD 8K
MEM MOD 8K
MEM MOD 8K
LEGEND - 1 - HIGH SPEED
2 - SERIAL DAT
3 - CONTROLLER
4 - MEMORY CON
5 - MEMORY INT
6 - DISC UNIT
7 - DISC CONT
8 - CARD READ
9 - LINE PRIN



LEGEND - 1 - HIGH SPEED DATA INTERFACE
2 - SERIAL DATA INTERFACE
3 - CONTROLLER
4 - MEMORY CONTROLLER
5 - MEMORY INTERFACE
6 - DISC UNIT
7 - DISC CONTROLLER
8 - CARD READER
9 - LINE PRINTER

FIGURE D-1. CONVENTIONAL COMPUTER SYSTEM CONFIGURATION

APPENDIX E

COMPUTER SYSTEM LIFE CYCLE COST MODEL

Ownership Cost Model - F-18 OFT Advanced Concept Computer System

1. Initial acquisition cost of processor hardware - C_{01}
2. Initial acquisition cost of memory hardware - C_{02}
3. Initial acquisition cost of interface hardware - C_{03}
4. Initial acquisition cost of peripheral electronics - C_{04}
5. Initial acquisition cost of peripheral hardware - C_{05}
6. Initial acquisition cost of processor hardware documentation - C_{06}
7. Initial acquisition cost of memory hardware documentation - C_{07}
8. Initial acquisition cost of interface hardware documentation - C_{08}
9. Initial acquisition cost of peripheral electronics documentation - C_{09}
10. Initial acquisition cost of peripheral hardware documentation - C_{10}
11. Initial acquisition cost of OFT simulation software - C_{11}
12. Initial acquisition cost of utility software - C_{12}
13. Initial acquisition cost of maintenance spares for processor, - C_{13} memory, and interface hardware.
14. Initial acquisition cost of maintenance spares for peripheral - C_{14} electronics and peripheral hardware.
15. Initial cost of maintenance training - C_{15}
16. Initial cost of SSA programmer orientation - C_{16}
17. Initial cost of test equipment - C_{17}
18. Life cycle maintenance cost (0-5 years) hardware - C_{18} (processor, memory, interface)
19. Life cycle maintenance costs (5-10 years) - hardware - C_{19} (processor, memory, interface)
20. Life cycle maintenance costs (0-5 years) - hardware personnel - C_{20}
21. Life cycle maintenance costs (5-10 years) - hardware personnel - C_{21}
22. Life cycle maintenance costs (0-5 years) - software personnel - C_{22}
23. Life cycle maintenance costs (5-10 years) - software personnel - C_{23}

24. Life cycle maintenance cost (0-5 years) - peripheral electronics - C_{24} hardware

25. Life cycle maintenance cost (5-10 years) - peripheral electronics - C_{25} hardware

26. Life cycle maintenance cost (0-5 years) - peripheral hardware - C_{26}

27. Life cycle maintenance cost (5-10 years) - peripheral hardware - C_{27}

28. Life cycle maintenance cost (0-5 years) - peripheral electronics - C_{28} hardware personnel

29. Life cycle maintenance cost (5-10 years) - peripheral electronics - C_{29}

30. Life cycle maintenance cost - SSA programmer re-orientation - C_{30}

APPENDIX F

CONVENTIONAL COMPUTER SYSTEM HARDWARE COST

COMPUTER SYSTEM HARDWARE COST THREE CPU CONFIGURATION

A. PROCESSOR HARDWARE COSTS

No.	Name	Unit Cost	Total Cost
3	CPU	21,000	63,000
3	System Control Panel	3,000	9,000
3	Hex Display	600	18,000
3	Interrupt/RT Clock	3,000	9,000
3	Floating Point Unit	25,000	75,000
2	Extra Logic Chassis	1,500	3,000
5	Logic Power Supply	1,500	7,500
3	AC Distribution Panel	1,000	3,000
2	Double CPU Cabinets	1,600	3,200
2	Bay Extender	300	600
		Subtotal Costs	$191,300

B. MEMORY HARDWARE COSTS

SLAVE (16K)

No.	Name	Unit Cost	Total Cost
2	Memory Module (8K)	6,300	12,600
1	Memory Chassis	1,000	1,000
1	Memory Power Supply	2,000	2,000
1	Memory Controller	3,500	3,500
1	Memory Interface	4,000	4,000
		Subtotal Costs	$23,100

SLAVE (16K)

2	Memory Modules (8K)	6,300	12,600
1	Memory Chassis	1,000	1,000
1	Memory Power Supply	2,000	2,000
1	Memory Controller	3,500	3,500
1	Memory Interface	4,000	4,000
		Subtotal Costs	$23,100

MASTER (48K)

1	Memory Package (32K)	17,000	17,000
2	Memory Modules (8K)	6,300	12,600
1	Memory Chassis	1,000	1,000
1	Memory Power Supply	2,000	2,000
1	Memory Controller	3,500	3,500
		Subtotal Costs	$36,100

SHARED (16K)

No.	Name	Costs	Total Cost
2	Memory Modules (8K)	6,300	12,600
1	Memory Chassis	1,000	1,000
1	Memory Power Supply	2,000	2,000
3	Memory Controller	3,500	10,500
		Subtotal Costs	$ 26,100
		Total Memory Cost	$108,400

C. INTERFACE ELECTRONICS

3	High Speed Interface	4,000	12,000
1	Serial Interface	3,500	3,500
1	Serial Data Interface	4,000	4,000
		Subtotal Costs	$19,500

D. PERIPHERAL ELECTRONICS

3	Controller	3,000	9,000
3	Disc Controller	5,000	15,000
3	Magnetic Tape Controller	3,500	10,500
1	Tape Formatter	2,500	2,500
		Subtotal Costs	$37,000

E. PERIPHERAL ELECTRONICS

No.	Name	Cost	Total Cost
3	Alphanumeric CRT	2,200	6,600
1	Card Reader	6,600	6,600
1	Line Printer	16,000	16,000
1	Moving HD Disc	15,000	15,000
1	Magnetic Tape Transport	15,000	15,000
1	Peripheral Cabinet	1,500	1,500
1	Peripheral Switch	20,000	20,000
		Subtotal costs	$80,700

COST SUMMARY		% OF TOTAL
A/B Processor & Memory Hardware	$299.7K	68.6
C. Interface Electronics	19.5K	4.4
D. Peripheral	37.0K	8.5
E. Peripheral Units	80.7K	18.5
Total Costs	$436.9K	100.0

APPENDIX G

LIFE CYCLE COST ANALYSIS FOR CONVENTIONAL COMPUTER SYSTEM APPROACH

Ownership Costs - F-18 OFT Advanced Concept Computer System

Three CPU Configuration

1. Initial acquisition costs of processor hardware - C_{01}

   C_{01} = \$191,300

2. Initial acquisition costs of memory hardware - C_{02}

   C_{02} = \$108,400

3. Initial acquisition cost of interface hardware - C_{03}

   C_{03} = \$19,500

4. Initial acquisition cost of peripheral electronics - C_{04}

   C_{04} = \$37,000

5. Initial acquisition cost of peripheral hardware - C_{05}

   C_{05} = \$80,700

6. Initial acquisition cost of processor hardware documentation - C_{06}

   (Cost of drawings, manuals, etc.)

CPU -	\$100.00
Real-time clock & Interrupt -	\$ 20.00
Turnkey Panel -	\$ 6.00
System Control Panel -	\$ 15.00
Logic Power Supply Set -	\$ 10.00
Total C_{06} =	\$151.00

7. Initial acquisition costs of memory hardware documentation - C_{07}

Memory Controller -	\$15.00
Mem. Mod (32KB) -	10.00
Memory Adaptor -	10.00
Total C_{07} =	\$35.00

8. Initial acquisition costs of interface hardware documentation - C_{08}

Serial Interface -	\$15.00
High Speed Interface -	30.00
Controller -	15.00
Disc Controller -	15.00
MTC -	15.00
Total C_{08} =	\$90.00

9. Initial acquisition cost of peripheral electronics documentation - C_{09}

MTF -	\$30.00
Total C_{09} -	\$30.00

10. Initial acquisition cost of peripheral hardware documenation - C_{10}

Alphanumeric CRT -	\$30.00
Card Reader -	25.00
Printer -	50.00
Magnetic Tape Unit -	30.00
Moving Head Disc Unit -	45.00
Total C_{10} -	\$180.00

11. Initial acquisition cost of OFT Simulation Software - C_{11}

Program size estimate - 38,000 instructions

Estimate 1 manhour per instruction

Estimated directed cost - \$8.60 per manhour (Per NTEC Proc. Dept Memo)

C_{11} = (38,000)(\$8.60) - \$326,800

12. Initial acquisition cost of utility software - C_{12}

Operating System	-	1500.00
MACRO ASSEMB	-	750.00
FORTRAN IV COMPILER	-	1500.00
MATH LIBRARY	-	250.00
SCIENTIFIC LIBRARY	-	250.00
DIAGNOSTIC PROGRAMS	-	250.00
MANUALS	-	340.00

Total C_{12} =\$4840.00 x 2 (for 2 copies)

=\$9680.00

13. Initial acquisition cost of maintenance spares (for memory, controllers, IOM's)

C_{13} - \$57,000

14. Initial acquisition cost of maintenance spares (for peripheral - C_{14} electronics and peripheral hardware)

Card Reader	-	\$3000.00
Line Printer	-	5000.00
Tape Unit	-	4000.00
Moving Head Disc	-	5000.00
Magnetic Tape Formatter	-	1500.00

Total C_{14} = \$18,500.00

15. Initial cost of maintenance training - C_{15}

Firmware (2 wks x $375/wk)	$ 750.00
Hardware (4 wks x $375/wk)	1500.00
Peripherals (3 wks x $375/wk)	1125.00
Personnel Costs (9 wk salary)	3465.00
Per Diem ($280/wk x 9 wk)	2520.00
Total C_{15}	$9360.00

16. Initial cost of SSA programmer orientation - C_{16}

Assembly language (1 wk x $3750/wk)	$ 375.00
FORTRAN (1wk x $375/wk)	375.00
Advanced (1 wk x $375/wk)	375.00
05 (2 wk x $375/wk)	750.00
Personnel Cost (5 wk salary)	1925.00
Per Diem ($280/wk x 5)	1400.00
Total C_{16}	$5200.00

17. Initial cost of test equipment - C_{17}

Maintenance panel and adapters	$10,000.00
Special tools (proc & mem)	50,000.00
Total C_{17}	$60,000.00

18. Life cycle maintenance costs (0-5 years) - hardware - C_{18}

(processors, memory, interfaces)

Computed as 10% of acquisition costs of spares ($57,000 per year)

0-5 years - (5 x $5700)

C_{18} = $28,500.00

19. Life cycle maintenance costs (5-10 years) - hardware - C_{19}
(processors, memory, interfaces)

5-10 years - (5 x $5700)

C_{19} = $28,500.00

20. Life cycle maintenance costs (0-5 years) - hardware personnel - C_{20}
Tech time - 1 manyear @ $20,000

5 x $20,000 = $100,000.00

C_{20} = $100,000.00

21. Life cycle maintenance cost (5-10 years) - hardware personnel - C_{21}

5 x $20,000 = $100,000.00

C_{21} = $100,000.00

22. Life cycle maintenance costs (0-5 years) - software personnel - C_{22}

1 full time engineer - $32,000/yr

5 x $32,000 = $160,000.00

C_{22} = $160,000.00

23. Life cycle maintenance costs (5-10 years) - software personnel - C_{23}

1 full time engineer - $32,000.00

5 x $32,000 = $160,000.00

C_{23} = $160,000.00

24. Life cycle maintenance costs (0-5 years) - peripheral elec. hardware - C_{24}

Computed as 6% of acquisition cost of spares ($18,500) per year

(.06) (18,500) = $1110

5 x $1110 = $5550

C_{24} = $5550.00

25. Life cycle maintenance cost (5-10 years) - peripheral elec. hardware - C_{25}

Computed as 6% of acquisition costs of spares ($18,500) per year

(.06) (18,500) = $1110

5 x $1110 = $5550

C_{25} = $5550.00

26. Life cycle maintenance costs (0-5 years) - peripheral hardware - C_{26}

Computed as 6% of acquisition costs of spares ($18,500) per year

(.06) (18,500) = $1110

5 x $1110 = $5550

C_{26} = $5550.00

27. Life cycle maintenance costs (5-10 years) - peripheral hardware - C_{27}

Computed as 6% of acquisition costs of spares ($18,500) per year

(.06) (18,500) = $1110

5 x $1110 = $5550

C_{27} = $5550.00

28. Life cycle maintenance cost (0-5 years) - peripheral electronics - C_{28} and hardware personnel

1 technician @ $20,000/manyear

5 x $20,000 = $100,000

C_{28} = $100,000.00

29. Life cycle maintenance cost (5-10 years) - peripheral electronics - C_{29} and hardware personnel

1 technician @ $20,000/manyear

5 x $20,000 - $100,000

C_{29} = $100,000.00

30. Life cycle maintenance costs - SSA Programmer reorientation - C_{30}

(New programmer - one time cost - 5th year)

Based on item 16 costs - assumed the same.

C_{30} = $5200.00

TOTAL GENERAL PURPOSE COMPUTER SYSTEM APPROACH

F-18 OFT CONVENTIONAL COMPUTER SYSTEM CONCEPT

C_{01} - Initial processor hardware	$191,300
C_{02} - Initial memory hardware	108,400
C_{03} - Initial interface hardware	19,500
C_{04} - Initial peripheral electronics	37,000
C_{05} - Initial peripheral hardware	80,700
C_{06} - Initial processor hdwe doc.	151
C_{07} - Initial memory hdwe doc.	35
C_{08} - Initial interface hdwe doc.	90
C_{09} - Initial peripheral elex. doc.	30
C_{10} - Initial peripheral hdwe doc.	180
C_{11} - Initial OFT simul. software	326,800
C_{12} - Initial utility software	9,680
C_{13} - Initial proc. mem. interface spares	57,000
C_{14} - Initial peripheral elex & hdwe sprs	18,500
C_{15} - Initial maintenance training	9,360
C_{16} - Initial SSA prog. orientation	5,200
C_{17} - Initial test equipment	60,00
C_{18} - L.C. (0-5 years) hdwe (P,M, I)	28,500
C_{19} - L.C. (5-10 years) hdwe (P,M,I)	28,500
C_{20} - L.C. (0-5 years) hdwe personnel	160,000
C_{21} - L.C. (5-10 years) hdwe personnel	160,000
C_{22} - L.C. (0-5 years) software	160,000

TOTAL GENERAL PURPOSE COMPUTER SYSTEM APPROACH

F-18 OFT CONVENTIONAL COMPUTER SYSTEM CONCEPT (CONT'D)

C_{23} - L.C. (0-5 years) software personnel	$ 160,000
C_{24} - L.C. (5-10 years) periph elex. hdwe maint.	5,550
C_{25} - L.C. (5-10 years) periph elex hdwe maint.	5,550
C_{26} - L.C. (0-5 years) periph hdwe maint.	5,550
C_{27} - L.C. (5-10 years) periph hdwe maint.	5,550
C_{28} - L.C. (0-5 years) per elex & hdwe personnel	100,000
C_{29} - L.C. (5-10 years) per elex & hdwe personnel	100,000
C_{30} - L.C. SSA Prog. reorientation	5,200
10 YEAR TOTAL COST OWNERSHIP	$1,728,326

APPENDIX H

MICROCOMPUTER PERFORMANCE ANALYSIS

TABLE H-1. MICROCOMPUTER PERFORMANCE ANALYSIS

MICROCOMPUTER TYPE: INTEL 3000 BIPOLAR SET CLOCK PERIOD: 100 nsec

INSTRUCTION TYPE	% USAGE	CLOCK CYCLES	MEMORY ADDRESS CYCLES	MEMORY ACCESS CYCLES	TOTAL CYCLES	CLOCK CYCLE TIME	INSTRUCTION EXECUTION TIME	WEIGHTED AVERAGE EXEC.
LOAD	.208	4	4	2	10	-	1.000	.209
STORE	.158	4	4	2	10	-	1.000	.158
ADD/SUBT	.092	9	4	2	15	-	1.500	.138
MULTIPLY	.058	41	2	1	44	-	4.400	.255
DIVIDE	.008	109	2	1	112	-	11.600	.093
LOGICAL	.068	7	4	2	13	-	1.300	.088
SHIFT (5 PL)	.020	14	2	1	17	-	1.700	.034
COMPARE	.047	7	4	2	13	-	1.300	.061
BRANCH	.161	8	4	2	14	-	1.400	.225
INDEX	.003	14	4	2	20	-	2.000	.006
REG-TO-REG	.031	6	-	-	6	-	0.600	.019
MISCELLANEOUS	.141	14	2	1	17	-	1.700	.240
I/O SET-UP + 6 WORDS	.004	4	3	3	10	-	1.000	.400
AVERAGE EXECUTION TIME (MICROSECONDS PER INSTRUCTION)								1.926
EQUIVALENT INSTRUCTIONS PER SECOND								519,211

TABLE H-2. MICROCOMPUTER PERFORMANCE ANALYSIS

MICROCOMPUTER TYPE: MC-10800 MECL FAMILY — CLOCK PERIOD: 50 nsec

INSTRUCTION TYPE	% USAGE	CLOCK CYCLES	MEMORY ADDRESS CYCLES	MEMORY ACCESS CYCLES	TOTAL CYCLES	CLOCK CYCLE TIME	INSTRUCTION EXECUTION TIME	WEIGHTED AVERAGE EXEC.
LOAD	.208	8	6	6	20	50	1.000	0.209
STORE	.158	8	6	6	20	50	1.000	0.158
ADD/SUBT	.092	8	6	6	20	50	1.000	0.092
MULTIPLY	.058	20	9	9	38	50	1.900	0.1102
DIVIDE	.008	24	9	9	42	50	2.100	0.0168
LOGICAL	.068	8	6	6	20	50	1.000	0.068
SHIFT (5 PL)	.020	8	11	11	30	50	1.500	0.030
COMPARE	.047	5	9	9	23	50	1.150	0.0541
BRANCH	.161	5	9	9	23	50	1.150	0.1852
INDEX	.003	8	8	6	20	50	1.000	0.003
REG-TO-REG	.031	3	3	3	9	50	.450	0.014
MISCELLANEOUS	.141	20	9	9	38	50	1.900	0.2679
I/O SET-UP + 6 WORDS	.004	8	8	9	25	50	1.250	0.005
AVERAGE EXECUTION TIME (MICROSECONDS PER INSTRUCTION)								1.2093
EQUIVALENT INSTRUCTIONS PER SECOND								826.925

TABLE H-3. MICROCOMPUTER PERFORMANCE ANALYSIS

MICROCOMPUTER TYPE: TI-SN74S481, SN74S482 ELEMENTS							CLOCK PERIOD: 100 nsec	
INSTRUCTION TYPE	% USAGE	CLOCK CYCLES	MEMORY ADDRESS CYCLES	MEMORY ACCESS CYCLES	TOTAL CYCLES	CLOCK CYCLE TIME	INSTRUCTION EXECUTION TIME	WEIGHTED AVERAGE EXEC.
LOAD	.208	2	8	-	10	100	1.1000	.2090
STORE	.158	2	8	-	10	100	1.1000	.1580
ADD/SUBT	.092	4	8	-	12	100	1.3000	.1196
MULTIPLY	.058	19	8	-	27	100	2.7000	.1506
DIVIDE	.008	23	8	-	31	100	3.1000	.0248
LOGICAL	.068	6	8	-	14	100	1.4000	.0952
SHIFT (5 PL)	.020	9	8	-	17	100	1.7000	.0340
COMPARE	.047	11	8	-	19	100	1.9000	.0893
BRANCH	.161	11	8	-	19	100	1.9000	.3059
INDEX	.003	15	16	-	31	100	3.1000	.0093
REG-TO-REG	.031	6	3	-	9	100	0.9000	.0279
MISCELLANEOUS	.141	12	8	-	20	100	2.0000	.2820
I/O SET-UP + 6 WORDS	.004	11	38	-	49	100	4.9000	.0196
AVERAGE EXECUTION TIME (MICROSECONDS PER INSTRUCTION)								1.5312
EQUIVALENT INSTRUCTIONS PER SECOND								653,083

TABLE H-4. MICROCOMPUTER PERFORMANCE ANALYSIS

MICROCOMPUTER TYPE: INTEL 8080A — CLOCK PERIOD: 2 usec

INSTRUCTION TYPE	% USAGE	CLOCK CYCLES	MEMORY ADDRESS CYCLES	MEMORY ACCESS CYCLES	TOTAL CYCLES	CLOCK CYCLE TIME	INSTRUCTION EXECUTION TIME	WEIGHTED AVERAGE EXEC.
LOAD	.208	7	1	1	9	2	18	3.744
STORE	.158	7	1	1	9	2	18	2.844
ADD/SUBT	.092	7	--	--	7	2	18	1.288
MULTIPLY	.058	252	1	1	254	2	508	29.464
DIVIDE	.008	262	1	1	266	2	528	4.224
LOGICAL	.068	7	--	--	7	2	14	0.952
SHIFT (5 PL)	.020	20	--	--	20	2	40	0.800
COMPARE	.047	7	1	1	9	2	18	0.846
BRANCH	.161	10	1	1	12	2	24	3.864
INDEX	.003	16	1	1	18	2	36	0.108
REG-TO-REG	.031	5	--	--	5	2	10	0.310
MISCELLANEOUS	.141	11	1	1	12	2	26	3.666
I/O SET-UP + 6 WORDS	.004	70	7	7	84	2	168	0.672
AVERAGE EXECUTION TIME (MICROSECONDS PER INSTRUCTION)								52.782
EQUIVALENT INSTRUCTIONS PER SECOND								47,590

TABLE H-5. MICROCOMPUTER PERFORMANCE ANALYSIS

MICROCOMPUTER TYPE: AM 2900 SERIES — CLOCK PERIOD: 100 nsec

INSTRUCTION TYPE	% USAGE	CLOCK CYCLES	MEMORY ADDRESS CYCLES	MEMORY ACCESS CYCLES	TOTAL CYCLES	CLOCK CYCLE TIME	INSTRUCTION EXECUTION TIME	WEIGHTED AVERAGE EXEC.
LOAD	.208	8	2	3	13	.100	1.3	0.2704
STORE	.158	8	2	3	13	.100	1.3	0.2054
ADD/SUBT	.092	8	2	3	13	.100	1.3	0.1196
MULTIPLY	.058	17	1	3	21	.100	2.1	0.1218
DIVIDE	.008	40	2	3	45	.100	4.5	0.0360
LOGICAL	.068	8	2	3	13	.100	1.3	0.0884
SHIFT (5 PL)	.020	14	1	3	18	.100	1.8	0.0360
COMPARE	.047	6	2	3	11	.100	1.1	0.0517
BRANCH	.161	8	2	3	13	.100	1.3	0.2093
INDEX	.003	14	2	3	19	.100	1.9	0.0057
REG-TO-REG	.031	3	-	-	3	.100	.3	0.0093
MISCELLANEOUS	.141	12	2	3	17	.100	1.7	0.2397
I/O SET-UP + 6 WORDS	.004	6	2	12	20	.100	2.0	0.0080
AVERAGE EXECUTION TIME (MICROSECONDS PER INSTRUCTION)								1.4013
EQUIVALENT INSTRUCTIONS PER SECOND								713,623

TABLE H-6. MICROCOMPUTER PERFORMANCE ANALYSIS

MICROCOMPUTER TYPE: FAIRCHILD 9440 MICRO FAMILY - 16 BIT BIPOLAR   CLOCK PERIOD: 200 nsec

INSTRUCTION TYPE	% USAGE	CLOCK CYCLES	MEMORY ADDRESS CYCLES	MEMORY ACCESS CYCLES	TOTAL CYCLES	CLOCK CYCLE TIME	INSTRUCTION EXECUTION TIME	WEIGHTED AVERAGE EXEC.
LOAD	.209						3.000	0.627
STORE	.158						4.500	0.711
ADD/SUBT	.092						1.500	0.138
MULTIPLY	.058						23.000	1.334
DIVIDE	.008						25.000	0.200
LOGICAL	.068						5.000	0.340
SHIFT (5 PL)	.020						11.300	0.266
COMPARE	.047						4.500	0.212
BRANCH	.161						7.500	1.208
INDEX	.003						7.500	0.023
REG-TO-REG	.031						1.500	0.047
MISCELLANEOUS	.141						4.500	0.635
I/O SET-UP + 6 WORDS	.004						2.500	0.010
AVERAGE EXECUTION TIME (MICROSECONDS PER INSTRUCTION)								5.711
EQUIVALENT INSTRUCTIONS PER SECOND								175,100

TABLE H-7. MICROCOMPUTER PERFORMANCE ANALYSIS

MICROCOMPUTER TYPE: INTEL 8085A-2							CLOCK PERIOD:	.8 usec
INSTRUCTION TYPE	% USAGE	CLOCK CYCLES	MEMORY ADDRESS CYCLES	MEMORY ACCESS CYCLES	TOTAL CYCLES	CLOCK CYCLE TIME	INSTRUCTION EXECUTION TIME	WEIGHTED AVERAGE EXEC.
LOAD	.208	7	1	1	9	.8	7.2	1.4976
STORE	.158	7	1	1	9	.8	7.2	1.1376
ADD/SUBT	.092	7	-	-	7	.8	5.6	0.5152
MULTIPLY	.058	252	1	1	254	.8	203.2	11.7856
DIVIDE	.008	262	1	1	266	.8	212.8	1.7024
LOGICAL	.068	7	-	-	7	.8	5.6	0.3808
SHIFT (5 PL)	.020	20	-	-	20	.8	16.0	0.32
COMPARE	.047	7	1	1	9	.8	7.2	0.3384
BRANCH	.161	10	1	1	12	.8	9.6	1.5456
INDEX	.003	16	1	1	18	.8	14.4	0.0432
REG-TO-REG	.031	5	-	-	5	.8	4.0	0.124
MISCELLANEOUS	.141	11	1	1	12	.8	9.6	1.3536
I/O SET-UP + 6 WORDS	.004	70	7	7	84	.8	67.2	0.2688
AVERAGE EXECUTION TIME (MICROSECONDS PER INSTRUCTION)								21.0128
EQUIVALENT INSTRUCTIONS PER SECOND								47,590

TABLE H-8. MICROCOMPUTER PERFORMANCE ANALYSIS

MICROCOMPUTER TYPE: TI-SPP-9900 CLOCK PERIOD: .3846 usec

INSTRUCTION TYPE	% USAGE	CLOCK CYCLES	MEMORY ADDRESS CYCLES	MEMORY ACCESS CYCLES	TOTAL CYCLES	CLOCK CYCLE TIME	INSTRUCTION EXECUTION TIME	WEIGHTED AVERAGE EXEC.
LOAD	.208	5	3	10	18	---	6.9228	1.4469
STORE	.158	60	4	10	74	---	28.4604	4.4967
ADD/SUBT	.092	28	8	10	56	---	21.5376	1.9814
MULTIPLY	.058	52	5	20	67	---	25.7682	1.4946
DIVIDE	.008	124	6	10	140	---	53.8440	.4308
LOGICAL	.068	14	4	10	28	---	10.7688	.7323
SHIFT (5 PL)	.020	52	4	10	66	---	25.3836	.5076
COMPARE	.047	--	-	--	--	---	---	--
BRANCH	.161	12	3	10	25	---	9.6150	1.5480
INDEX	.003	14	4	10	28	---	10.7688	.0323
REG-TO-REG	.031	--	-	--	--	---	---	--
MISCELLANEOUS	.141	12	4	10	26	---	9.9996	1.4099
I/O SET-UP + 6 WORDS	.004	112	7	20	139	---	53.4594	.2138
AVERAGE EXECUTION TIME (MICROSECONDS PER INSTRUCTION)								14.2943
EQUIVALENT INSTRUCTIONS PER SECOND								69,958

APPENDIX I

PROCESSING AND STORAGE REQUIREMENTS PER MICROCOMPUTER BASED ON F-18 OFT CONCEPT REQUIREMENTS

MICROCOMPUTER MODULE ASSIGNMENT AND DEFINITION

MICROCOMPUTER #1
MODULE #1 (30 HZ) - INSTRUCTOR STATION,INSTRUCTION STATION DISPLAYS, BRAKE FORCES,NOSE WHEEL STEERING,STRUT DEFLECTION, RATES,GROUND TURNING RATES,VERTICAL STRUT FORCES, LONG/LAT. GEAR FORCES,TOTAL GEAR FORCES,TOTAL GEAR MOMENTS, GROUND CONTACT CHECK.

MICROCOMPUTER #2
MODULE #2 (60 HZ) - AERO, FORCE,MOMENT COMP STABILITY AXIS,ATMOSPHERE, WEIGHT AND BALANCE.

MICROCOMPUTER #3
MODULE #3 (60 HZ) - FLIGHT CONTROL SYSTEM, FLIGHT CONTROL/CPU INTERFACE, PROPULSION SYSTEM.

MICROCOMPUTER #4
MODULE #4 (60 HZ) - STABILITY AXIS-BODY AXIS TRANSFORMATION,BODY-TO-EARTH TRANSFORMATION,EARTH AXIS ACCEL. INTEGRATION, EARTH AXIS VELOCITY INTEGRATIONS, EARTH-BODY AXIS TRANSFORMATION,ROTATIONAL ACCEL.-P,Q,R DOT, MOMEMTS-BODY AXIS,P,Q,R INTEGRATION,AIR DATA COMPUTATIONS.

MICROCOMPUTER #5
MODULE #5 (60 HZ) - ANGULAR POSITIONS (QUATERNIONS),G-SEAT & G-SUIT, TACAN MATH MODEL.

MICROCOMPUTER #6
MODULE #6 (30 HZ) - NAVIGATION DATA COMP.,RADAR NAV.FACILITIES SIM., COMPASS SIM.,COCKPIT INSTRUMENTS,UHF-VHF SIM., DATA LINK-ILS SIM.,RADAR ALTIMETER SIM., ACCESSORIES SIM., ON-BOARD COMPUTER INTERFACE.

TABLE I-1. PROCESSING AND STORAGE REQUIREMENTS

MICROCOMPUTER & MODULE ASSIGNMENT	INSTRUCTION ESTIMATES	CONSTANT & DATA ESTIMATES	TOTAL MEMORY REQUIRED	WORST-CASE INSTRUCTION EXECUTION RATE	AVERAGE INSTRUCTION EXECUTION TIME (usec)
MICROCOMPUTER #1 MODULE 1 (30HZ)	6,784	331	7,115	211,157	4.7358
MICROCOMPUTER #2 MODULE 2 (60HZ)	4,725	600	5,325	384,750	2.5991
MICROCOMPUTER #3 MODULE 3 (60HZ)	4,556	651	5,206	331,594	3.0157
MICROCOMPUTER #4 MODULE 4 (60HZ)	2,717	413	3,129	226,167	4.4215
MICROCOMPUTER #5 MODULE 5 (60HZ)	3,459	288	3,747	359,438	2.7821
MICROCOMPUTER #6 MODULE 6 (30HZ)	7,298	1,125	8,423	303,497	3.2949
SYSTEM TOTALS	29,539	3,408	32,945	1,816,603	------

AVERAGE INSTRUCTION EXECUTION TIME PER SYSTEM................. 0.5505 MICROSECONDS

APPENDIX J

FRAME TIME ANALYSIS - MICROCOMPUTER APPROACH

TABLE J-1. FRAME TIME ANALYSIS - MICROCOMPUTER APPROACH

MICROCOMPUTER & MODULE	FRAME RATE (per sec)	FRAME TIME (msec)	FRAME ASSIGNMENTS
MICROCOMPUTER #1 AND MODULE #1	30	8.851	1 THROUGH 30
MICROCOMPUTER #2 AND MODULE #2	60	8.064	1 THROUGH 60
MICROCOMPUTER #3 AND MODULE #3	60	6.950	1 THROUGH 60
MICROCOMPUTER #4 AND MODULE #4	60	4.740	1 THROUGH 60
MICROCOMPUTER #5 AND MODULE #5	60	7.533	1 THROUGH 60
MICROCOMPUTER #6 AND MODULE #6	30	12.722	1 THROUGH 30

APPENDIX K

MICROCOMPUTER HARDWARE COST

MICROCOMPUTER SYSTEM COST ESTIMATES

32 BIT DEVICE (USING AMD-2900 SERIES)

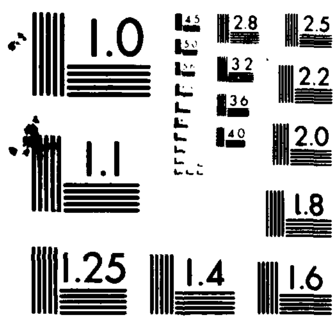
* Assuming 1 Microcomputer System is Manufactured	Cost-Each QTY-10-99	Quantity Required	Total Cost
Processor & Memory Hardware			
Material Cost (Components)	800	10	4,000
Memory Costs (8K Each)	2000	10	20,000
Power Supplies	700	4	2,800
Power Supply (Mem)	600	1	600
Card Cages	1000	2	2,000
Racks	1000	1	1,000
Cabling	500	1	500
			24,900
Design Labor - 1.5 MY @ 25K			37,500
MFG Labor - 1.5 MY @ 15K			22,500
Total Labor			60,000
Overhead - Materials 50%			17,450
Overhead - Labor 60%			36,000
Total Overhead			53,450
Subtotal			148,350
G & A - 12%			17,802
Subtotal			166,152
Profit - 7%			11,630
Total Per Unit Cost (EST)			$177,782

INTERFACE ELECTRONICS

NO.	NAME	UNIT COST	TOTAL COST
3	High Speed Data Interface	$4,000	$12,000
1	Serial Data Interface	4,000	4,000

AD-A108 744 NAVAL TRAINING EQUIPMENT CENTER ORLANDO FL F/G 9/2
AN ANALYSIS OF MICROCOMPUTER TECHNOLOGY FOR APPLICATION TO REAL--ETC(U)
JUL 79 C F SUMMER, G A WYNDLE
UNCLASSIFIED NAVTRAEQUIPC-IH-313 NL

END
DATE
FILMED
1 82
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963 A

			$16,000

PERIPHERAL ELECTRONICS

1	Peripheral Controller (Multiple Unit)	$3,000	$ 3,000
1	Moving HD Disc Controller	5,000	5,000

			$ 8,000

PERIPHERAL HARDWARE

1	CRT Terminal	$ 3,500	$ 3,500
1	Line Printer	4,000	4,000
1	Moving HD Disc	15,000	15,000
1	Peripheral Cabinent	1,500	1,500

			$24,000

COST SUMMARY

	Quantity	% of Total
Processor Hardware	177,782	79
Interface Electronics	16,000	7
Peripheral Electronics	8,000	4
Peripheral Units	24,000	10
	-------	----
	$225,782	100

APPENDIX L

LIFE CYCLE COST ANALYSIS FOR MICROCOMPUTER SYSTEM

OWNERSHIP COST - ADVANCED OFT CONCEPT - COMPUTER SYSTEM

MICROCOMPUTER APPROACH

C_{01} & C_{02} Initial acquisition cost of processor & memory hardware

C_{01} & C_{02} = \$177,782

C_{03} Initial acquisition cost of interface hardware

C_{03} = \$16,000

C_{04} Initial acquisition cost of peripheral electronics

C_{04} = \$8,000

C_{05} Initial acquisition cost of peripheral hardware

C_{05} = \$24,000

C_{06} Initial acquisition cost of processor hardware documentation

Microcomputer System	\$30
Logic Power Supply	10
Misc	30
	\$70

C_{06} = \$70

C_{07} Initial acquisition cost of memory hardware documentation

Memory System - \$25

C_{08} Initial acquisition cost of interface hardware documentation

SDI	-	\$15
HDI	-	30
TLC	-	15
CDC	-	15
		\$75

C_{08} = \$75

C_{09} Initial acquisition cost of peripheral electronics documentation

C_{09} = \$30

C_{10} Initial acquisition cost of peripheral hardware documentation

CRT	-	$ 30
Printer	-	50
Moving HD Disc	-	45
		$125

C_{10} = $125

C_{11} Initial acquisition cost of OFT simulation software

Program size estimate - (38,000 - 14,000) instructions
Estimate 1 manhour per instruction
Estimated direct cost - $8.60 per manhour (per N-6)

C_{11} = (26,000) ($8.60) = $223,600
Standardization of the flight math model is achieved with this approach. This will reduce the number of instructions to be coded by 14,000.

C_{12} Initial acquisition cost of utility software

Macro Assemb	-	$400
Math Library	-	150
Scientific Lib	-	150
Manuals	-	200
		$900 x 2 (for 2 copies) = $1800

C_{12} = $1800

C_{13} Initial acquisition cost of maintenance spares (for microcomputer memory and the like)

Microcomputer	1060 x 4 = $ 4,240
Memory	2000 x 4 = 8,000
Power Supply	650 x 2 = 1,300
	$13,540

C_{13} = $14,540

C14 Initial acquisition cost of maintenance spares (for peripheral electronics and peripheral hardware)

Line printer	$5,000
Cartridge Disc	3,000
	$8,000

C_{14} = $8,000

C_{15} Initial cost of maintenance training

Firmware (3 wks x $375/wk)	$1125
Peripherals (3 wks x $375/wk)	1125
Personnel Cost (6 wks salary)	2310
Per Diem ($280/wk x 6 wks)	1680
	$6240

C_{16} Initial cost of SSA programmer orientation

Assembly Language (1 wk x $375/wk)	$ 375
System training (2 wk x $375/wk)	750
Personnel Cost (3 wk salary)	1155
Per Diem ($280/wk x 3)	840
	$3120

C_{17} Initial cost of test equipment

MDS - $30,000

C_{18} Life cycle maintenance costs (0-5 years) - hardware (processor, memory, interfaces). Computed as 10% of acquisition costs of spares ($13,540) per year.

0 - 5 yrs - 5 x $1354
C_{18} = $6770

C_{19} Life cycle maintenance costs (5-10 years) - hardware (processor, memory, interfaces)

5 - 10 yrs - 5 x $1354
C_{19} = $6770

C_{20} Life cycle maintenance costs (0-5 years) - hardware personnel

Tech. time - ¼ man year @ $20,000/manyear
5 x $5,000 = $25,000

C_{20} = $25,000

C_{21} Life cycle maintenance costs (5-10 years) - hardware personnel

5 x $5,000 = $25,000

C_{21} = $25,000

C_{22} Life cycle maintenance costs (0-5 years) - software personnel

1 full time engineer - \$32,000/yr
1/2 time - \$16,000/yr Time shared between two training systems
5 x \$16,000 = \$80,000

C_{22} = \$80,000

C_{23} Life cycle maintenance costs (5-10 years) - software personnel

1 full time engineer - \$32,000/yr
5 x \$16,000 = \$80,000

C_{23} = \$80,000

C_{24} Life cycle maintenance costs (0-5 years) - peripheral elec hardware

Computed as 6% acquisition cost of spares (\$8000) per year
(.06) (\$8000) = \$480
5 x \$480 = \$2400

C_{24} = \$2400

C_{25} Life cycle maintenance costs (5-10 years) - peripheral elec hardware

Computed as 6% of acquisition cost of spares (\$8000) per year
(.06) (\$8000) = \$480

5 x \$480 = \$2400

C_{25} = \$2400

C_{26} Life cycle maintenance cost (0-5 years) - peripheral hardware

Computed as 6% of acquisition cost of spares (\$8000) per year
(.06) (\$8000) = \$480
5 x \$480 = \$2400

C_{27} Life cycle maintenance costs (5-10 years) - peripheral hardware

Computed as 6% of acquisition cost of spares (\$8000) per year
(.06) (\$8000) = \$480
5 x \$480 = \$2400

C_{28} Life cycle maintenance costs (0-5 years) + (5-10 years) - peripheral electronics

C_{29} Tech time - 3/4 manyear @ \$20,000/manyear
5 x \$15,000 + 5 x \$15,000 = \$150,000

C_{30} Life cycle maintenance costs - SAA Programmer reorientation

(New programmer - one time cost 5th year)

Based upon standardization of this approach, SSA programmer reorientation consists of familiarizing the programmer with system particularities.

APPENDIX M

LIFE CYCLE COST COMPARISON

LIFE CYCLE COST COMPARISON

F-18 COMPUTER SYSTEM LIFE CYCLE COST ANALYSIS
COMPARISON AND SUMMARY

I. Computer Hardware System Cost

	CONVENTIONAL	MICROCOMPUTER
a. Hardware C_1 - C_5	436,900	225,782
b. Documentation C_6-C_{10} &C_{12}	10,166	2,125
I. Total	$447,066	$227,907

Δ 219K - 49%

II. Trainer Computer Acquisition & Development Cost

	CONVENTIONAL	MICROCOMPUTER
a. Simulation software development effort		
C_{11}	326,800	223,600
b. Initial Computer System Spare		
C_{13}	15,000	13,540
C_{14}	18,500	8,000
c. Maintenance training		
C_{15}	9,360	6,240
C_{16}	5,200	3,120
d. Test equipment		
C_{17}	60,000	30,000
II. Total	$476,860	$284,500

Δ 192K - 40%

III. Ten year support

a. Computer hardware spares

C_{18}	24,500	6,770
C_{19}	28,500	6,770

b. Hardware personnel

C_{20}	100,000	25,000
C_{21}	100,000	25,000

c. Software personnel

C_{22}	160,000	80,000
C_{23}	160,000	80,000

d. Peripheral elec. hardware

C_{24}	5,550	2,400
C_{25}	5,550	2,400

e. Peripheral hardware

C_{26}	5,550	2,400
C_{27}	5,550	2,400

f. Peripheral & electronic maint. personnel

C_{28}	100,000	75,000
C_{29}	100,000	75,000

g. SSA programmer reorientation

C_{30}	5,200	1,310
III. Total	$804,400	$384,450

420K - 52%

Total Life Cycle Cost $1,728,326 - $896,857

Δ 831K - 48%

DISTRIBUTION LIST

Naval Training Equipment Center (50)
Orlando, FL 32813

Defense Technical Info Center (12)
Cameron Station
Alexandria, VA 22314

University of South Carolina (1)
College of Engineering (Dr Pettus)
Electrical and Computer Engineering
Columbia, SC 29208

PM TRADE, US ARMY (2)
Attn: SE
Orlando, FL 32813

ALL OTHERS RECEIVE ONE COPY

Chief of Naval Education & Training
Code N-331
Pensacola, FL 32508

Naval Air Systems Command
AIR 340D
Washington, DC 20350

Naval Air Systems Command
AIR 340F
Washington, DC 20350

Naval Air Systems Command
AIR 360B, Attn: B. Zempolich
Washington, DC 20350

Naval Air Systems Command
Library AIR 50174
Washington, DC 20350

Chief of Naval Material, Navy Dept
Code MAT 08T
Washington, DC 20360

Chief of Naval Material, Navy Dept
MAT 08Y
Washington, DC 20360

Chief of Naval Material, Navy Dept
MAT 09Y
Washington, DC 20360

Chief of Naval Research
ONR 461
800 North Quincy St
Arlington, VA 22217

Naval Air Systems Command
AIR 413
Washington, DC 20350

AF ASD/ENE
Attn: Mr A. Doty
Wright-Patterson AFB, OH 45433

AF ASD/ENO
Attn: Mr Phil Babel
Wright-Patterson AFB, OH 45433

Dept of Electrical & Computer Engineering
Attn: Dr Harold Stone
University of Massachusetts
Amherst, MA 01003

Chief of Naval Operations
OP-115
Attn: LCDR J. Lawhon
Washington, DC 20350

DA

FILM